Chapter 6

Direct Verification: Proving a Property

In this chapter we illustrate techniques to prove the validity of a probabilistic statement from scratch. The main technique, which is based on *coin lemmas*, consists of reducing the analysis of a property of a probabilistic automaton to the analysis of a property of an ordinary automaton. We illustrate the methodology by applying it to some existing randomized algorithms.

Part of this chapter is based on joint work with Anna Pogosyants and Isaac Saias. Anna Pogosyants suggested us the coin event *OCC* (Section 6.2.3) as a generalization of other less elegant coin events that we had in mind and collaborated on the verification of the randomized algorithm for agreement of Ben-Or (Section 6.5). The verification of the randomized dining philosophers algorithm of Lehmann and Rabin (Section 6.3) is based on joint work with Nancy Lynch and Isaac Saias [LSS94], and the verification of the randomized algorithm for agreement of Ben-Or is a formalization of a proof that appears in the book on distributed algorithms of Nancy Lynch [Lyn95].

6.1 How to Prove the Validity of a Probabilistic Statement

In Chapter 5 we have defined formally what is a probabilistic statement and we have shown how it is possible to combine probabilistic statements to derive more complex properties. However, one question is left open: how do we prove the validity of a given probabilistic statement from scratch?

The problem is not trivial: a property may rely on complicate global configurations of a system that depend on several separated random draws. Analyzing the exact probability of an event associated with a probabilistic execution fragment may be extremely hard. Fortunately, there are usually some key points, known to the designer of a system, where specific probabilistic choices lead to the desired property. In this chapter we formalize the idea above by introducing a collection of *coin lemmas*. The idea behind a coin lemma is the following.

1. We define a mechanism to identify events of the kind "some specific probabilistic choices yield some specific results". We call such events *coin events* since a common source of randomness is given by coin flips.

2. We prove a lower bound on the probability of the coin event that we identify.

Then, the analysis of a probabilistic statement for a probabilistic automaton M proceeds as follows.

1. We find a coin event that expresses the key intuition behind the property to be shown.
2. We show that the coin event is a subevent of the event expressing the desired property, i.e., we show that whenever the coin event is satisfied, the desired property is satisfied as well.
3. We use the lower bound on the probability of the coin event to obtain a lower bound on the probability of the desired property.

Example 6.1.1 (Coin lemmas and the toy resource allocation protocol) Let us consider the toy resource allocation protocol of Chapter 5 again. One of the coin lemmas of this chapter states that if we fix any two separate coin flips (flipping of different coins) and we consider the event where the two coin flips yield different outcomes whenever they both occur, then, no matter how the nondeterminism is resolved, the considered event is satisfied with probability at least 1/2. On the other hand, if the first coin flip of M_1 after the first coin flip of M_2 is different from the last coin flip of M_2 before the first time M_1 checks its resource after flipping, then M_1 succeeds in getting its resource. Thus, whenever the property above can be expressed as a *coin event* in a form suitable to the coin lemma above, we are guaranteed that M_1 eventually gets its resource with probability at least 1/2. It turns out that an adversary must be fair, oblivious and deterministic in order to be able to define the desired coin event (cf. Section 6.6). Our results about deterministic and randomized adversaries (Proposition 5.7.11) can then be used to remove the constraint that an adversary is deterministic. ■

We present a large collection of coin lemmas, and we illustrate their use via two main examples: Section 6.3 proves the correctness of the randomized Dining Philosophers algorithm of Lehmann and Rabin [LR81], and Section 6.5 proves the correctness of the randomized algorithm of Ben-Or for agreement in asynchronous networks in the presence of stopping faults [BO83]. At the end of the chapter we hint at another technique, called the *partition technique*, that departs considerably from the coin lemmas and that is necessary to prove stronger claims about the toy resource allocation protocol. We leave to further work a deeper study of this other technique.

6.2 Some Simple Coin Lemmas

In this section we present some simple coin lemmas where we use actions to identify the random draws of interest. Specifically, we study the following coin lemmas.

1. *First occurrence of an action.*

   In this coin lemma we consider an action a and a set of states U, and we study the probability that either action a does not occur or the first occurrence of action a leads to a state of U. We show that this probability is at least the infimum of the probability of reaching a state of U over all the transitions of M that are labeled with action a.

As an example, action a can identify the process of flipping a fair coin and U can identify those states that are reached if the coin flip yields head. Then the coin lemma says that no matter how the nondeterminism is resolved the probability that either the coin is not flipped or the coin is flipped and yields head is at least 1/2.

Observe that in the definition of the coin event we allow for those executions where no coin is flipped. One reason for this choice is to avoid trivial lower bounds due to the fact that a generic adversary can always decide not to schedule any transition. Another reason is that generally a randomized algorithm is structured so that that if no coin is flipped then progress is guaranteed with certainty. Alternatively, a randomized algorithm can be structured so that under any valid adversary some coin is flipped. In both cases it is of absolute importance to be aware of the existence of executions where no coin is flipped. Overlooking those executions is a common source of mistakes.

2. *First occurrence of an action among many.*

   In this coin lemma we consider several pairs (a_i, U_i) of actions and sets of states, and we study the probability that either none of the a_i's occur or the action a_j that occurs first leads to a state of U_j. We show that, if for each i p_i is the lower bound given for (a_i, U_i) by the coin lemma of 1, then the probability mentioned above is at least the minimum of the p_i's.

   As an example, consider n processes that run in parallel, and suppose that each process can flip a fair coin. Then, the probability that either no process flips a coin or that the first process that flips a coin obtains head is at least 1/2.

3. *I-th occurrence of an action among many.*

   In this coin lemma we consider the coin event of 2 with the difference that we consider the i^{th} occurrence of an action rather than the first occurrence. The lower bound on the probability of this event is the same as the lower bound on the probability of the event of 2.

4. *Conjunction of separate coin events.*

   In this coin lemma we consider the conjunction of several coin events of the kind of 3. We show that if each one of the coin events involves disjoint occurrences of actions, then the lower bound on the probability of the conjunction is the product of the lower bounds on the probability of each of the involved coin events.

   As an example, consider n processes that run in parallel, and suppose that each process can flip a fair coin. For each i let x_i be either head or tail. Then, the probability that for each process i either no coin is flipped or the first coin that is flipped yields x_i is at least $1/2^n$.

Some more general and complex coin lemmas are presented in Section 6.4; several other coin lemmas are likely to be derived in the future. Before presenting the simple coin lemmas in full detail we give just a rough idea of the coin lemmas of Section 6.4.

5. *Conjunction of separate coin events with multiple outcomes.*

In this coin lemma we consider again the conjunction of several coin events that involve disjoint occurrences of actions. However we allow more freedom. First of all an action is paired with more than one set of states, thus allowing the observation of more than one outcome; second, we allow for multiple joint observations.

As an example, the coin lemma says that if n processes run in parallel and each one of them can flip a coin, then the probability that at least half of the processes either do not flip a coin or flip head is at least 1/2. Similarly, if each process can roll a dice, then the probability that if process 1 rolls 1 then the other processes do not roll a number different from 1 is at least $(1/6)^n + 5/6$, which is essentially the probability of rolling n dices and that either all processes give 1 or process 1 does not give 1.

6. *A generalized coin lemma.*

   In this coin lemma we generalize the idea of 5, but this time we do not use actions to identify the random draws of interest. The reader is referred to Section 6.4.2 for further details.

6.2.1 First Occurrence of an Action

Let M be a probabilistic automaton, and let (a, U) be a pair consisting of an action of M and a set of states of M. Let $FIRST(a, U)$ be a function that applied to a probabilistic execution fragment H of M returns the set of executions α of Ω_H such that either a does not occur in $\alpha \triangleright q_0^H$, or a occurs in $\alpha \triangleright q_0^H$ and the state reached after the first occurrence of a is a state of U.

It is simple to check that $FIRST(a, U)$ is an event schema since, for each probabilistic execution fragment H of M, the complement of $FIRST(a, U)(H)$ is the set of executions α of Ω_H such that action a occurs in $\alpha \triangleright q_0^H$, and the state reached after the first occurrence of a is not a state of U. This set is expressible as a union of cones, and thus it is an event.

The event schema $FIRST(a, U)$ identifies the first random draw associated with action a that occurs in a probabilistic execution fragment H, and requires the outcome of the random draw to be in a specific range, namely in U. The intuition behind the use of such a coin event, is that a system performs well if the outcome of the first random draw involving a is in U. From the definition of $FIRST(a, U)$, we assume also that the system performs well whenever a does not occur at all. Thus, if an adversary has the possibility not to schedule a, then it has a better chance to degrade the performance of a system by scheduling a.

The following lemma provides a lower bound to the probability of $FIRST(a, U)$. Informally, it states that if whenever there is a transition of M that involves action a the occurrence of a implies that a state of U is reached with probability at least p, then p is a lower bound on the probability of $FIRST(a, U)$.

Lemma 6.2.1 *Let M be a probabilistic automaton, and let (a, U) be a pair consisting of an action of M and a set of states of M. Let p be a real number between 0 and 1 such that for each transition $(s, \mathcal{P})$ of M where $P[a] > 0$, $P[U|a] \geq p$. Then, for each probabilistic execution fragment H of M, $P_H[FIRST(a, U)(H)] \geq p$.*

Proof. For convenience denote $FIRST(a, U)(H)$ by E, and for each state q of H, denote by $\Omega(q, \overline{U})$ the set $\{(a, q') \in \Omega_q^H \mid lstate(q') \notin U\}$. Let Θ be the set of states q of H such that action a does not occur in $q \triangleright q_0^H$, and $P_q^H[a] > 0$. Then,

$$P_H[\overline{E}] = \sum_{q \in \Theta} \sum_{(a,q') \in \Omega(q,\overline{U})} P_H[C_q] P_q^H[(a,q')]. \tag{6.1}$$

By expressing $P_q^H[(a,q')]$ as a conditional probability and rearranging the expression, we obtain

$$P_H[\overline{E}] = \sum_{q \in \Theta} P_H[C_q] P_q^H[a] \left(\sum_{(a,q') \in \Omega(q,\overline{U})} P_q^H[(a,q')|a] \right). \tag{6.2}$$

From the definition of a probabilistic execution fragment and the definition of $\Omega(q,\overline{U})$, for each element q of Θ there is a combined transition $tr = \sum_i p_i tr_i$ of M such that $tr_q^H = q \frown tr$ and

$$\sum_{(a,q') \in \Omega(q,\overline{U})} P_q^H[(a,q')|a] = P_{tr}[\overline{U}|a] = \frac{P_{tr}[\overline{U} \cap a]}{P_{tr}[a]} = \frac{\sum_i p_i P_{tr_i}[\overline{U} \cap a]}{\sum_i p_i P_{tr_i}[a]}. \tag{6.3}$$

By multiplying and dividing each i^{th} summand of the enumerator by $P_{tr_i}[a]$, using the hypothesis of the lemma, i.e., for each i $P_{tr_i}[\overline{U} \cap a] \leq (1-p)$, and simplifying algebraically, from (6.3) we obtain

$$\sum_{(a,q') \in \Omega(q,\overline{U})} P_q^H[(a,q')|a] \leq (1-p). \tag{6.4}$$

By using (6.4) in (6.2) we obtain

$$P_H[\overline{E}] \leq (1-p) \left(\sum_{q \in \Theta} P_H[C_q] P_q^H[a] \right). \tag{6.5}$$

Furthermore, the subexpression $\sum_{q \in \Theta} P_H[C_q] P_q^H[a]$ is the probability that a occurs in H, which is at most 1. Thus,

$$P_H[\overline{E}] \leq (1-p). \tag{6.6}$$

This completes the proof. ■

6.2.2 First Occurrence of an Action among Many

The event schema $FIRST(a,U)$ can be generalized to account for the first action that occurs among several possible ones. Let M be a probabilistic automaton, and let $(a_1,U_1),\ldots,(a_n,U_n)$ be pairs consisting of an action of M and a set of states of M such that the actions a_i are all distinct. Then define $FIRST((a_1,U_1),\ldots,(a_n,U_n))$ to be the function that applied to a probabilistic execution fragment H of M returns the set of executions α of Ω_H such that either none of the a_i's occurs in $\alpha \triangleright q_0^H$, or some of the a_i's occur in $\alpha \triangleright q_0^H$, and if a_i is the first of those actions that occurs, then the state reached after the first occurrence of a_i is a state of U_i.

It is simple again to check that $FIRST((a_1,U_1),\ldots,(a_n,U_n))$ is an event schema since, for each probabilistic execution fragment H, the complement of $FIRST((a_1,U_1),\ldots,(a_n,U_n))(H)$ can be expressed as a union of cones.

Lemma 6.2.1 extends to this case.

Lemma 6.2.2 *Let M be a probabilistic automaton, and let $(a_1, U_1), \ldots, (a_n, U_n)$ be pairs consisting of an action of M and a set of states of M such that the actions a_i are all distinct. Let $\{p_i\}_{i=1,\ldots,n}$ be a collection of real numbers between 0 and 1 such that for each i, $1 \leq i \leq n$, and each transition $(s, \mathcal{P})$ of M where $P[a_i] > 0$, $P[U|a_i] \geq p_i$. Then, for each probabilistic execution fragment H of M, $P_H[\mathit{FIRST}((a_1, U_1), \ldots, (a_n, U_n))(H)] \geq \mathit{min}(p_1, \ldots, p_n)$.*

Proof. Let V be $\{a_1, \ldots, a_n\}$, and let p be the minimum of $\{p_1, \ldots, p_n\}$. For convenience, denote $\mathit{FIRST}((a_1, U_1), \ldots, (a_n, U_n))(H)$ by E, and for each state q of H, denote by $\Omega(q, \overline{E})$ the set $\cup_{i \in \{1,\ldots,n\}}\{(a_i, q') \in \Omega_q^H \mid \mathit{lstate}(q') \notin U_i\}$. Then, for each transition $(q, \mathcal{P}_q^H)$ of H such that $P_q^H[V] > 0$,

$$P_q^H[\Omega(q, \overline{E})|V] \leq (1 - p). \tag{6.7}$$

To prove (6.7), let, for each $i = 1, \ldots, n$, $\Omega(q, a_i, \overline{U}_i)$ denote the set $\{(a_i, q') \in \Omega_q^H \mid \mathit{lstate}(q') \notin U_i\}$. Then,

$$P_q^H[\Omega(q, \overline{E})|V] = \sum_{i \in \{1,\ldots,n\}} P_q^H[\Omega(q, a_i, \overline{U}_i)|V]. \tag{6.8}$$

By using conditional probabilities, Equation (6.8) can be rewritten into

$$P_q^H[\Omega(q, \overline{E})|V] = \sum_{i \in \{1,\ldots,n\}} P_q^H[a_i|V] P_q^H[\Omega(q, a_i, \overline{U}_i)|a_i]. \tag{6.9}$$

Following the same argument as in the proof of Lemma 6.2.1, for each i, $P_q^H[\Omega(q, a_i, \overline{U}_i)|a_i] \leq (1 - p)$; moreover, $\sum_i P_q^H[a_i|V] = 1$. Thus, (6.7) follows directly.

The rest of the proof follows te lines of the proof of Lemma 6.2.1. Let Θ be the set of states q of H such that no action of V occurs in $q \rhd q_0^H$, and $P_q^H[V] > 0$. Then,

$$P_H[\overline{E}] = \sum_{q \in \Theta} \sum_{(a,q') \in \Omega(q,\overline{E})} P_H[C_q] P_q^H[(a, q')]. \tag{6.10}$$

By expressing $P_q^H[(a, q')]$ as a conditional probability and rearranging the expression, we obtain

$$P_H[\overline{E}] = \sum_{q \in \Theta} P_H[C_q] P_q^H[V] \left(\sum_{(a,q') \in \Omega(q,\overline{E})} P_q^H[(a, q')|V] \right). \tag{6.11}$$

The subexpression $\sum_{(a,q') \in \Omega(q,\overline{E})} P_q^H[(a, q')|V]$ is $P_q^H[\Omega(q, \overline{E})|V]$, which is less than or equal to $(1 - p)$ from (6.7). Thus,

$$P_H[\overline{E}] \leq (1 - p) \left(\sum_{q \in \Theta} P_H[C_q] P_q^H[V] \right). \tag{6.12}$$

Furthermore, the subexpression $\sum_{q \in \Theta} P_H[C_q] P_q^H[V]$ is the probability that an action from V occurs in H, which is at most 1. Thus,

$$P_H[\overline{E}] \leq (1 - p). \tag{6.13}$$

This completes the proof. ■

6.2.3 I-th Occurrence of an Action among Many

In the definition of *FIRST* we have considered the first action among a given set that occurs in a probabilistic execution fragment H. However, the results for *FIRST* are valid also if we consider the i^{th} occurrence of an action instead of the first occurrence. This observation suggests a new more general event schema.

Let M be a probabilistic automaton, and let $(a_1, U_1), \ldots, (a_n, U_n)$ be pairs consisting of an action of M and a set of states of M such that the actions a_i are all distinct. Then define $OCC(i, (a_1, U_1), \ldots, (a_n, U_n))$ to be the function that applied to a probabilistic execution fragment H of M returns the set of executions α of Ω_H such that either there are less than i occurrences of actions from $\{a_1, \ldots, a_n\}$ in $\alpha \triangleright q_0^H$, or there are at least i occurrences of actions from $\{a_1, \ldots, a_n\}$, and, if a_j is the action that occurs as the i^{th} one, then the state reached after its occurrence is a state of U_i.

Since in the proof of Lemma 6.2.2 we never use the fact that it is the first occurrence of an action that is considered, Lemma 6.2.2 carries over to the i^{th} occurrence trivially.

Lemma 6.2.3 *Let M be a probabilistic automaton, and let $(a_1, U_1), \ldots, (a_n, U_n)$ be pairs consisting of an action of M and a set of states of M such that the actions a_i are all distinct. Let $\{p_j\}_{j=1,\ldots,n}$ be a collection of real numbers between 0 and 1 such that for each $j \in \{1, \ldots, n\}$ and each transition $(s, \mathcal{P})$ of M where $P[a_j] > 0$, $P[U|a_j] \geq p_j$. Then, for each probabilistic execution fragment H of M, $P_H[OCC(i, (a_1, U_1), \ldots, (a_n, U_n))(H)] \geq min(p_1, \ldots, p_n)$.* ■

6.2.4 Conjunction of Separate Coin Events

In this section we study what happens if we consider several events of the kind OCC together. In order to simplify the notation, we consider only event schemas of the kind $OCC(i, (a, U))$ since, as we have seen in the proof of Lemma 6.2.2, the case with multiple actions can be reduced to the case with a single action.

The lemma that we prove states that if we consider several separate coin events, i.e., coin events that involve different random draws, each one with its own lower bound, then the lower bound of their conjunction is the product of the lower bounds. In other words, an adversary can introduce dependencies by increasing the probability of the conjunction of events, but it can never decrease the probability below the value that we would get by considering all the events to be independent.

Lemma 6.2.4 *Let M be a probabilistic automaton, and let $(k_1, a_1, U_1), \ldots, (k_n, a_n, U_n)$ be a collection of triplets consisting of a natural number, an action of M and a set of states of M, such that the pairs (k_i, a_i) are all distinct. Let $\{p_j\}_{j=1,\ldots,n}$ be a collection of real numbers between 0 and 1 such that for each $j \in \{1, \ldots, n\}$ and each transition $(s, \mathcal{P})$ of M where $P[a_j] > 0$, $P[U|a_j] \geq p_j$. Then, for each probabilistic execution fragment H of M, $P_H[OCC(k_1, (a_1, U_1))(H) \cap \cdots \cap OCC(k_n, (a_n, U_n))(H)] \geq p_1 \cdots p_n$.*

Proof. For each $I \subseteq \{1, \ldots, n\}$, denote a generic event schema $\cap_{i \in I} OCC(k_i, (a_i, U_i))$ by e_I. For each $i = 1, \ldots, n$ and each state q of H, denote by $\Omega(q, i, U_i)$ the set $\{(a_i, q') \in \Omega_q^H \mid \mathit{lstate}(q') \in U_i\}$ of pairs where a_i occurs and U_i is reached, and denote by $\Omega(q, i, \overline{U}_i)$ the set $\{(a_i, q') \in \Omega_q^H \mid \mathit{lstate}(q') \notin U_i\}$ of pairs where a_i occurs and U_i is not reached. For each action

a and each state q of H, let $a(q)$ denote the number of occurrences of action a in $q \triangleright q_0^H$. For each $i = 1, \ldots, n$, let Θ_i be the set of states q of H such that each action $a_j, 1 \leq j \leq n$ occurs less than k_j times in $q \triangleright q_0^H$, action a_i occurs $k_i - 1$ times in $q \triangleright q_0^H$, and $P_q^H[a_i] > 0$. For each $i = 1, \ldots, n$ and each state q of H such that $a_i(q) < k_i$, let $OCC(k_i, (a_i, U_i)) \triangleright q$ denote the event schema $OCC(k_i - a_i(q), (a_i, U_i))$. Finally, for each $I \subseteq \{1, \ldots, n\}$ and each suitable state q of H, let $e_I \triangleright q$ denote the event schema $\cap_{i \in I} OCC(k_i, (a_i, U_i)) \triangleright q$.

We prove the lemma by induction on n. If $n = 1$, then the result follows directly from Lemma 6.2.1. Otherwise,

$$P_H[\overline{e_{1,\ldots,n}(H)}] = \sum_{i \in \{1,\ldots,n\}} \sum_{q \in \Theta_i} P_H[C_q] \left(\left(\sum_{(a_i,q') \in \Omega(q,i,\overline{U}_i)} P_q^H[(a_i, q')] \right) \right.$$
$$\left. + \left(\sum_{(a_i,q') \in \Omega(q,i,U_i)} P_q^H[(a_i, q')] P_{H \triangleright q'}[\overline{e_{\{1,\ldots,i-1,i+1,\ldots,n\}} \triangleright q'(H \triangleright q')}] \right) \right). \tag{6.14}$$

The first summand of Expression (6.14) expresses the probability that action a_i occurs from q and leads to a state not in U_i; the second summand expresses the probability that a_i occurs, leads to a state of U_i, and from the reached state something happen so that the resulting execution is not in $e_{1,\ldots,n}(H)$. From induction, and by using conditional probabilities, we obtain

$$P_H[\overline{e_{1,\ldots,n}(H)}] \leq \sum_{i \in \{1,\ldots,n\}} \sum_{q \in \Theta_i} P_H[C_q] P_q^H[a_i] \left(\left(\sum_{(a_i,q') \in \Omega(q,i,\overline{U}_i)} P_q^H[(a_i, q') | a_i] \right) \right.$$
$$\left. + \left(\sum_{(a_i,q') \in \Omega(q,i,U_i)} P_q^H[(a_i, q') | a_i])(1 - p_1 \cdots p_{i-1} p_{i+1} \cdots p_n) \right) \right). \tag{6.15}$$

Let, for each i and each q, $p_{i,q} = P_q^H[\Omega(q, i, U_i) | a_i]$. Then, (6.15) becomes

$$P_H[\overline{e_{1,\ldots,n}(H)}]$$
$$\leq \sum_{i \in \{1,\ldots,n\}} \sum_{q \in \Theta_i} P_H[C_q] P_q^H[a_i]((1 - p_{i,q}) + (1 - p_1 \cdots p_{i-1} p_{i+1} \cdots p_n) p_{i,q}), \tag{6.16}$$

which becomes

$$P_H[\overline{e_{1,\ldots,n}(H)}] \leq \sum_{i \in \{1,\ldots,n\}} \sum_{q \in \Theta_i} P_H[C_q] P_q^H[a_i](1 - p_1 \cdots p_{i-1} p_{i,q} p_{i+1} \cdots p_n) \tag{6.17}$$

after simple algebraic simplifications. Using the same argument as in the proof of Lemma 6.2.1, for each i and each q, $p_{i,q} \geq p_i$. Thus,

$$P_H[\overline{e_{1,\ldots,n}(H)}] \leq \sum_{i \in \{1,\ldots,n\}} \sum_{q \in \Theta_i} P_H[C_q] P_q^H[a_i](1 - p_1 \cdots p_n). \tag{6.18}$$

Finally, observe that $\sum_{i \in \{1,\ldots,n\}} \sum_{q \in \Theta_i} P_H[C_q] P_q^H[a_i]$ is the probability that for some i action a_i occurs at least k_i times. Thus,

$$P_H[\overline{e_{1,\ldots,n}(H)}] \leq (1 - p_1 \cdots p_n). \tag{6.19}$$

This completes the proof. ■

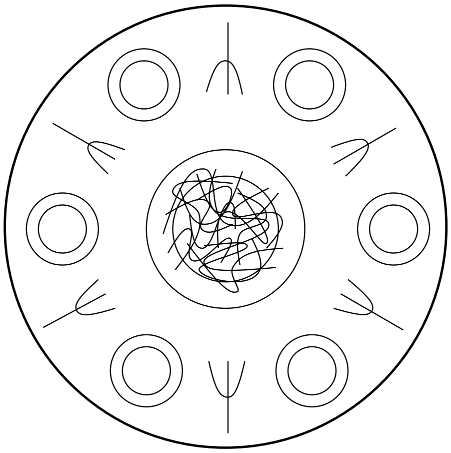


Figure 6-1: The Dining Philosopher problem with 6 philosophers.

6.3 Example: Randomized Dining Philosophers

In this section we apply the methodology presented so far to prove the correctness of the Randomized Dining Philosophers algorithm of Lehmann and Rabin [LR81]. The proof is structured in two levels. The high level proof consists of a collection of progress statements that are concatenated together; the low level proof consists of the proofs of the statements of the high level proof. The low level proof is based on the coin lemmas.

6.3.1 The Problem

There are n philosophers sat at a round table. Each philosopher has a plate in from of him, a fork on its left, and a fork on its right. The left fork is shared with his left neighbor philosopher, and the right fork is shared with his right neighbor philosopher. At the center of the table there is a bowl full of spaghetti. Figure 6-1 illustrates the situation for $n = 6$. Each philosopher goes repeatedly through phases where he is thinking and where he is eating. However, each philosopher needs both of its forks in order to eat. The problem is the following:

> "*What procedure should each philosopher follow to get his forks and to put them down in order to make sure that every philosopher that is hungry will eventually be able to eat?*"

A simpler problem is the following.

> "*What procedure should each philosopher follow to get his forks and to put them down in order to make sure that whenever somebody is hungry somebody will eventually be able to eat?*"

The second problem is simpler than the first problem since it allows for some philosopher to starve. It is known from [LR81] that there is no symmetric solution even for the simple dining philosophers problem, i.e., there is no deterministic solution for the dining philosophers problem where each philosopher follows exactly the same protocol; some mechanism to break the symmetry is necessary. In the algorithm of Lehmann and Rabin each philosopher follows exactly the same protocol and randomness is used to break the symmetry.

Shared variables: $\mathrm{Res}_j \in \{\mathtt{free}, \mathtt{taken}\}$, $j = 1, \ldots, n$, initially `free`.

Local variables: $u_i \in \{\mathtt{left}, \mathtt{right}\}$, $i = 1, \ldots, n$

Code for process i:

```
0.   try                                              ** beginning of Trying Section **
1.   < $u_i \leftarrow$ random>           ** choose left or right with equal probability **
2.   < if $\mathrm{Res}_{(i,u_i)}$ = free then
             $\mathrm{Res}_{(i,u_i)}$ := taken                  ** pick up first resource **
       else goto 2. >
3.    < if $\mathrm{Res}_{(i,opp(u_i))}$ = free then
             $\mathrm{Res}_{(i,opp(u_i))}$ := taken;            ** pick up second resource **
             goto 5. >
4.    < $\mathrm{Res}_{(i,u_i)}$ := free; goto 1.>               ** put down first resource **
5.    crit                                                  ** end of Trying Section **
      ** Critical Section **
6.    exit                                            ** beginning of Exit Section **
7.    < $u_i \leftarrow$ left or right                       ** nondeterministic choice **
        $\mathrm{Res}_{(i,opp(u_i))}$ := free >                ** put down first resources **
8.    < $\mathrm{Res}_{(i,u_i)}$ := free >                   ** put down second resources **
9.    rem                                                     ** end of Exit Section **
      ** Remainder Section **
```

Figure 6-2: The Lehmann-Rabin algorithm. The operations between angular brackets are performed atomically.

6.3.2 The Algorithm

Each hungry philosopher proceeds according to the following protocol.

1. Flip a fair coin to choose between the `left` and the `right` fork.
2. Wait for the chosen fork to become free and get it.
3. Try to get the second fork:
   if it is free, then get it;
   if it is taken, then put down the first fork and go to 1.
4. Eat.

Each philosopher that has terminated to eat puts down his forks one at a time. The intuition behind the use of randomness is that the actual protocol used by each philosopher is determined by an infinite sequence of random coin flips. Thus, with probability 1 each philosopher follows a different protocol.

Figure 6-2 gives a more precise representation of the protocol, using a terminology that is closer to computer science; thus, a philosopher is called a process, and a fork is called a resource. A philosopher who is thinking is said to be in its *reminder* region; a philosopher

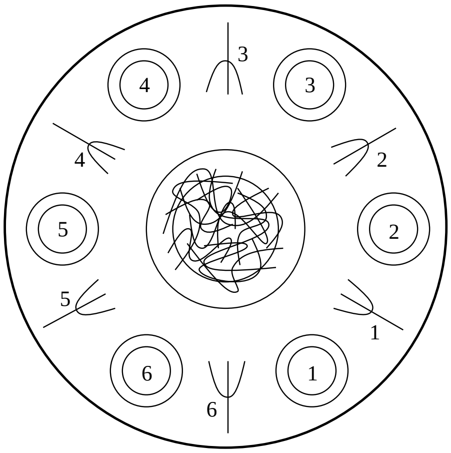


Figure 6-3: Numbering processes and resources in the Dining Philosophers problem.

who is eating is said to be in its *critical* region; a philosopher who is trying to get its forks is said to be in its *trying* region; and a philosopher who is putting down its forks is said to be in its *exit* region. The n resources (forks) are represented by n shared variables $\mathrm{Res}_1, \ldots, \mathrm{Res}_n$, each of which can assume values in $\{\mathtt{free}, \mathtt{taken}\}$. Each process (philosopher) i ignores its own name and the names of its adjacent resources. However, each process i is able to refer to its adjacent resources by relative names: $\mathrm{Res}_{(i,\mathtt{left})}$ is the resource located to the left, and $\mathrm{Res}_{(i,\mathtt{right})}$ is the resource to the right of i. Each process i has a private variable u_i, whose value is in $\{\mathtt{left}, \mathtt{right}\}$, which is used either to keep track of the resource that process i currently holds, or, if no resource is held, to keep track of the resource that process i is going to take next. For notational convenience we define an operator *opp* that complements the value of its argument, i.e., $\mathit{opp}(\mathtt{right}) = \mathtt{left}$ and $\mathit{opp}(\mathtt{left}) = \mathtt{right}$.

We now define a probabilistic automaton M that represents the evolution of n philosophers. We assume that process $i+1$ is on the right of process i and that resource Res_i is between processes i and $i+1$ (see Figure 6-3). We also identify labels modulo n so that, for instance, process $n+1$ coincides with process 1.

A state s of M is a tuple $(X_1, \ldots, X_n, \mathrm{Res}_1, \ldots, \mathrm{Res}_n)$ containing the local state X_i of each process i, and the value of each resource Res_i. Each local state X_i is a pair (pc_i, u_i) consisting of a program counter pc_i and the local variable u_i. The program counter of each process keeps track of the current instruction in the code of Figure 6-2. Rather than representing the value of the program counter with a number, we use a more suggestive notation which is explained in Table 6.1. Also, the execution of each instruction is represented by an action. Actions $\mathtt{try}_i$, $\mathtt{crit}_i$, $\mathtt{rem}_i$, $\mathtt{exit}_i$ are external; all the other actions are internal.

The start state of M assigns the value $\mathtt{free}$ to all the shared variables Res_i, the value R to each program counter pc_i, and an arbitrary value to each variable u_i. The transition relation of M is derived directly from Figure 6-2. For example, for each state where $pc_i = F$ there is an internal transition labeled with $\mathtt{flip}_i$ that changes pc_i into W and assigns $\mathtt{left}$ to u_i with probability 1/2 and $\mathtt{right}$ to u_i with probability 1/2; from each state where $X_i = (W, \mathtt{left})$ there is a transition labeled with $\mathtt{wait}_i$ that does not change the state if $\mathrm{Res}_{(i,\mathtt{left})} = \mathtt{taken}$, and changes pc_i into S and $\mathrm{Res}_{(i,\mathtt{left})}$ into $\mathtt{taken}$ if $\mathrm{Res}_{(i,\mathtt{left})} = \mathtt{free}$; for each state where

Nr.	pc_i	Action	Informal meaning
0	R	$\texttt{try}_i$	**R**eminder region
1	F	$\texttt{flip}_i$	Ready to **F**lip
2	W	$\texttt{wait}_i$	**W**aiting for first resource
3	S	$\texttt{second}_i$	Checking for **S**econd resource
4	D	$\texttt{drop}_i$	**D**ropping first resource
5	P	$\texttt{crit}_i$	**P**re-critical region
6	C	$\texttt{exit}_i$	**C**ritical region
7	E_F	$\texttt{dropf}_i$	**E**xit: drop **F**irst resource
8	E_S	$\texttt{drops}_i$	**E**xit: drop **S**econd resource
9	E_R	$\texttt{rem}_i$	**E**xit: move to **R**eminder region

Table 6.1: Program counter and action names for the Lehmann-Rabin algorithm.

$pc_i = E_F$ there are two transitions labeled with action $\texttt{dropf}_i$: one transition sets u_i to `right` and makes $\text{Res}_{(i,\texttt{left})}$ free, and the other transition sets u_i to `left` makes $\text{Res}_{(i,\texttt{right})}$ free. The two separate transitions correspond to a nondeterministic choice that is left to the adversary.

The value of each pair X_i can be represented concisely by the value of pc_i and an arrow (to the left or to the right) which describes the value of u_i. Thus, informally, a process i is in state $\underset{\rightarrow}{S}$ or $\underset{\rightarrow}{D}$ (resp. $\underset{\leftarrow}{S}$ or $\underset{\leftarrow}{D}$) when i is in state S or D while holding its right (resp. left) resource; process i is in state $\underset{\rightarrow}{W}$ (resp. $\underset{\leftarrow}{W}$) when i is waiting for its right (resp. left) resource to become free; process i is in state $\underset{\rightarrow}{E_S}$ (resp. $\underset{\leftarrow}{E_S}$) when i is in its exit region and it is still holding its right (resp. left) resource. Sometimes we are interested in sets of pairs; for example, whenever $pc_i = F$ the value of u_i is irrelevant. With the simple value of pc_i we denote the set of the two pairs $\{(pc_i, \texttt{left}), (pc_i, \texttt{right})\}$. Finally, with the symbol # we denote any pair where $pc_i \in \{W, S, D\}$. The arrow notation is used as before.

For each state $s = (X_1, \ldots, X_n, \text{Res}_1, \ldots, \text{Res}_n)$ of M we denote X_i by $X_i(s)$ and Res_i by $\text{Res}_i(s)$. Also, for any set St of states of a process i, we denote by $X_i \in St$, or alternatively $X_i = St$ the set of states s of M such that $X_i(s) \in St$. Sometimes we abuse notation in the sense that we write expressions like $X_i \in \{F, D\}$ with the meaning $X_i \in F \cup D$. Finally, we write $X_i = E$ for $X_i = \{E_F, E_S, E_R\}$, and we write $X_i = T$ for $X_i \in \{F, W, S, D, P\}$.

6.3.3 The High Level Proof

In this section we give the high level proof that the algorithm of Lehmann and Rabin guarantees progress, i.e., that from every state where some process is in its trying region, some process enters eventually its critical region with probability 1. We assume that each process that is ready to perform a transition is allowed eventually to do so: process i is ready to perform a transition whenever it enables an action different from $\texttt{try}_i$ or $\texttt{exit}_i$. Actions $\texttt{try}_i$ and $\texttt{exit}_i$ are under the control of the user (a philosopher decides whether to eat or think), and hence, by assumption, under the control of the adversary.

Formally, consider the probabilistic automaton M of Section 6.3.2. Define an extended execution α of M to be *fair* iff for each process i either α is finite and its last state enables

$\mathtt{try}_i$ or $\mathtt{exit}_i$, or α is infinite and either actions of process i occur infinitely many times in α or $\alpha = \alpha_1 \frown \alpha_2$ and all the states of α_2 enable either $\mathtt{try}_i$ or $\mathtt{exit}_i$. Define *Fairadvs* to be the set of adversaries $\mathcal{A}$ for M such that, for every finite execution fragment α of M the elements of $\Omega_{prexec(M,\mathcal{A},\alpha)}$ are extended fair execution fragments of M. Then *Fairadvs* is finite-history-insensitive: if $\mathcal{A}$ is an adversary of *Fairadvs* and q is a finite execution fragment of M, then it is easy to verify that the adversary $\mathcal{A}_q$ such that

$$\mathcal{A}_q(\alpha) = \begin{cases} \mathcal{A}(\alpha \rhd q) & \text{if } q \leq \alpha \\ \mathcal{A}(\alpha) & \text{otherwise} \end{cases}$$

is an adversary of *Fairadvs*. Let $rstates(M)$ denote the set of reachable states of M. Let

$$\mathcal{T} \triangleq \{s \in rstates(M) \mid \exists_i X_i(s) \in \{T\}\}$$

denote the sets of reachable states of M where some process is in its trying region, and let

$$\mathcal{C} \triangleq \{s \in rstates(M) \mid \exists_i X_i(s) = C\}$$

denote the sets of reachable states of M where some process is in its critical region. We first show that

$$\mathcal{T} \underset{1/8}{\longrightarrow}_{\mathit{Fairadvs}} \mathcal{C}, \tag{6.20}$$

i.e., that, starting from any reachable state where some process is in its trying region, for all the adversaries of *Fairadvs*, some process enters its critical region eventually with probability at least 1/8. Note that (6.20) is satisfied trivially if some process is initially in its critical region.

Our proof is divided into several phases, each one concerned with the property of making some partial progress toward $\mathcal{C}$. The sets of states associated with the different phases are expressed in terms of $\mathcal{T}, \mathcal{RT}, \mathcal{F}, \mathcal{G}, \mathcal{P}$, and $\mathcal{C}$. Here,

$$\mathcal{RT} \triangleq \{s \in \mathcal{T} \mid \forall_i X_i(s) \in \{E_R, R, T\}\}$$

is the set of states where at least one process is in its trying region and where no process is in its critical region or holds resources while being in its exit region.

$$\mathcal{F} \triangleq \{s \in \mathcal{RT} \mid \exists_i X_i(s) = F\}$$

is the set of states of $\mathcal{RT}$ where some process is ready to flip a coin.

$$\mathcal{P} \triangleq \{s \in rstates(M) \mid \exists_i X_i(s) = P\}$$

is the sets of reachable states of M where some process is in its pre-critical region, i.e., where some process is ready to enter its critical region. The set $\mathcal{G}$ is the most important for the analysis. To motivate the definition, we define the following notions. We say that a process i is *committed* if $X_i \in \{W, S\}$, and that a process i *potentially controls* Res_i (resp. Res_{i-1}) if $X_i \in \{\underrightarrow{W}, \underrightarrow{S}, \underrightarrow{D}\}$ (resp. $X_i \in \{\underleftarrow{W}, \underleftarrow{S}, \underleftarrow{D}\}$). Informally said, a state in $\mathcal{RT}$ is in $\mathcal{G}$ if and only if there is a committed process whose second resource is not potentially controlled by another process. Such a process is called a *good* process. Formally,

$$\begin{aligned} \mathcal{G} \triangleq \{s \in \mathcal{RT} \mid \exists_i \\ & X_i(s) \in \{\underleftarrow{W}, \underleftarrow{S}\} \text{ and } X_{i+1}(s) \in \{E_R, R, F, \underrightarrow{\#}\}, \text{ or} \\ & X_i(s) \in \{\underrightarrow{W}, \underrightarrow{S}\} \text{ and } X_{i-1}(s) \in \{E_R, R, F, \underleftarrow{\#}\}\} \end{aligned}$$

Reaching a state of $\mathcal{G}$ is a substantial progress toward reaching a state of $\mathcal{C}$. Somehow, a good state is a place where the symmetry is broken. The progress statements of the proof are the following.

$$\begin{array}{ll}
\mathcal{T} \xrightarrow[1]{} \mathcal{RT} \cup \mathcal{C} & \text{(Proposition 6.3.3)}, \\
\mathcal{RT} \xrightarrow[1]{} \mathcal{F} \cup \mathcal{G} \cup \mathcal{P} & \text{(Proposition 6.3.16)}, \\
\mathcal{F} \xrightarrow[1/2]{} \mathcal{G} \cup \mathcal{P} & \text{(Proposition 6.3.15)}, \\
\mathcal{G} \xrightarrow[1/4]{} \mathcal{P} & \text{(Proposition 6.3.12)}, \\
\mathcal{P} \xrightarrow[1]{} \mathcal{C} & \text{(Proposition 6.3.1)}.
\end{array}$$

The first statement says that eventually every process in its exit region relinquishes its resources. In this way we avoid to deal with resources held by processes who do not want to enter the critical region. The second statement says that eventually either a good state is reached, or a place where some process is ready to flip its coin is reached. The flipping points are potential points where the symmetry is broken, and thus reaching a flipping point means progress. The third statement says that from a flipping point there is probability 1/2 to reach a good state. Finally, the fourth statement says that from a good state there is probability 1/4 to be ready to enter the critical region. By combining the statements above by means of Proposition 5.5.3 and Theorem 5.5.2 we obtain

$$\mathcal{T} \xrightarrow[1/8]{} \mathcal{C}, \tag{6.21}$$

which is the property that was to be proven. Observe that once some process is in the trying region there is always some process in the trying region until some process reaches the critical region. Formally, M satisfies $\mathcal{T}$ *Unless* $\mathcal{C}$. Thus, Proposition 5.5.6 applies, leading to

$$\mathcal{T} \xrightarrow[1]{} \mathcal{C}. \tag{6.22}$$

6.3.4 The Low Level Proof

In this section we prove the five progress statements used in Section 6.3.3. The proofs are detailed operational arguments. The main point to observe is that randomness is handled exclusively by the coin lemmas, and thus, any technique for the verification of ordinary automata could be applied as well.

For the sake of clarity, we do not prove the relations in the order they were presented. Throughout the proof we abuse notation by writing expressions of the kind $\mathit{FIRST}(\mathtt{flip}_i, \mathtt{left})$ for the event schema $\mathit{FIRST}(\mathtt{flip}_i, \{s \in \mathit{states}(M) \mid X_i(s) = \underleftarrow{W}\})$. We write also sentences of the form "If $\mathit{FIRST}(\mathtt{flip}_i, \mathtt{left})$ then ϕ" meaning that for each valid probabilistic execution fragment H, each element of $\mathit{FIRST}(\mathtt{flip}_i, \mathtt{left})(H)$ satisfies ϕ.

Proposition 6.3.1 *If some process is in P, then some process enters C, i.e.,*

$$\mathcal{P} \xrightarrow[1]{} \mathcal{C}.$$

Proof. Let i be the process in P. Then, from the definition of *Fairadvs*, process i is scheduled eventually, and enters C. ■

Lemma 6.3.2 *If some process is in its Exit region, then it will eventually enter R.*

Proof. The process needs to perform two transitions to relinquish its two resources, and then one transition to send a `rem` message to the user. Every adversary of *Fairadvs* guarantees that those three transitions are performed eventually. ■

Proposition 6.3.3 $\mathcal{T} \longrightarrow \mathcal{RT} \cup \mathcal{C}$.

Proof. From Lemma 6.3.2, every process that begins in E_F or E_S relinquishes its resources. If no process begins in C or enters C in the meantime, then the state reached at this point is a state of $\mathcal{RT}$; otherwise, the starting state or the state reached when the first process enters C is a state of $\mathcal{C}$. ■

We now turn to the proof of $\mathcal{G} \xrightarrow[1/4]{} \mathcal{P}$. The following lemmas form a detailed cases analysis of the different situations that can arise in states of $\mathcal{G}$. Informally, each lemma shows that a specific coin event is a sub-event of the properties of reaching some other state. A preliminary lemma is an invariant of M, which guarantees that the resources are held by those processes who think to be holding them.

Lemma 6.3.4 *For each reachable state s of M and each i, $1 \leq i \leq n$, Res_i =* `taken` *iff $X_i(s) \in \{\underrightarrow{S}, \underrightarrow{D}, P, C, E_F, \underrightarrow{E_S}\}$ or $X_{i+1}(s) \in \{\underleftarrow{S}, \underleftarrow{D}, P, C, E_F, \underleftarrow{E_S}\}$. Moreover, for each reachable state s of M and each i, $1 \leq i \leq n$, it is not the case that $X_i(s) \in \{\underrightarrow{S}, \underrightarrow{D}, P, C, E_F, \underrightarrow{E_S}\}$ and $X_{i+1}(s) \in \{\underleftarrow{S}, \underleftarrow{D}, P, C, E_F, \underleftarrow{E_S}\}$, i.e., only one process at a time can hold one resource.* ■

Proof. The proof of this lemma is a standard proof of invariants. Simply verify that the two properties are true for the start states of M and are preserved by each transition of M. ■

Lemma 6.3.5

1. *Let $X_{i-1} \in \{E_R, R, F\}$ and $X_i = \underleftarrow{W}$. If FIRST(*$\texttt{flip}_{i-1}$*,* `left`*), then, eventually, either $X_{i-1} = P$ or $X_i = S$.*

2. *Let $X_{i-1} = D$ and $X_i = \underleftarrow{W}$. If FIRST(*$\texttt{flip}_{i-1}$*,* `left`*), then, eventually, either $X_{i-1} = P$ or $X_i = S$.*

3. *Let $X_{i-1} = S$ and $X_i = \underleftarrow{W}$. If FIRST(*$\texttt{flip}_{i-1}$*,* `left`*), then, eventually, either $X_{i-1} = P$ or $X_i = S$.*

4. *Let $X_{i-1} = W$ and $X_i = \underleftarrow{W}$. If FIRST(*$\texttt{flip}_{i-1}$*,* `left`*), then, eventually, either $X_{i-1} = P$ or $X_i = S$.*

Proof. The four proofs start in the same way. Let s be a state of M satisfying the respective properties of items *1* or *2* or *3* or *4*. Let $\mathcal{A}$ be an adversary of *Fairadvs*, and let α be an execution of $\Omega_{prexec(M,\{s\},\mathcal{A})}$ where the result of the first coin flip of process $i-1$, if it occurs, is `left`.

1. By hypothesis and Lemma 6.3.4, $i-1$ does not hold any resource at the beginning of α and has to obtain Res_{i-2} (its left resource) before pursuing Res_{i-1}. From the definition of *Fairadvs*, i performs a transition eventually in α. If $i-1$ does not hold Res_{i-1} when i performs this transition, then i progresses into configuration S. If not, it must be the case that $i-1$ succeeded in getting it in the meanwhile. But, in this case, since $i-1$ flips `left`, Res_{i-1} was the second resource needed by $i-1$ and $i-1$ therefore entered P.

2. If $X_i = S$ eventually, then we are done. Otherwise, process $i-1$ performs a transition eventually. Let $\alpha = \alpha_1 \frown \alpha_2$ such that the last transition of α_1 is the first transition taken by process $i-1$. Then $X_{i-1}(\mathit{fstate}(\alpha_2)) = F$ and $X_i(\mathit{fstate}(\alpha_2)) = \underleftarrow{W}$. Since process $i-1$ did not flip any coin during α_1, from the finite-history-insensitivity of *Fairadvs* and Item *1* we conclude.

3. If $X_i = S$ eventually, then we are done. Otherwise, process $i-1$ performs a transition eventually. Let $\alpha = \alpha_1 \frown \alpha_2$ such that the last transition of α_1 is the first transition taken by process $i-1$. If $X_{i-1}(\mathit{fstate}(\alpha_2)) = P$ then we are also done. Otherwise it must be the case that $X_{i-1}(\mathit{fstate}(\alpha_2)) = D$ and $X_i(\mathit{fstate}(\alpha_2)) = \underleftarrow{W}$. Since process $i-1$ did not flip any coin during α_1, from the finite-history-insensitivity of *Fairadvs* and Item *2* we conclude.

4. If $X_i = S$ eventually, then we are done. Otherwise, process i checks its left resource eventually and fails, process $i-1$ gets its right resource before, and hence reaches at least state S. Let $\alpha = \alpha_1 \frown \alpha_2$ where the last transition of α_1 is the first transition of α that leads process $i-1$ to state S. Then $X_{i-1}(\mathit{fstate}(\alpha_2)) = S$ and $X_i(\mathit{fstate}(\alpha_2)) = \underleftarrow{W}$. Since process $i-1$ did not flip any coin during α_1, from the finite-history-insensitivity of *Fairadvs* and Item *3* we conclude. ■

Lemma 6.3.6 *Assume that* $X_{i-1} \in \{E_R, R, T\}$ *and* $X_i = \underleftarrow{W}$. *If* $\mathit{FIRST}(\mathtt{flip}_{i-1}, \mathtt{left})$, *then, eventually, either* $X_{i-1} = P$ *or* $X_i = S$.

Proof. Follows directly from Lemma 6.3.5 after observing that $X_{i-1} \in \{E_R, R, T\}$ is equivalent to $X_{i-1} \in \{E_R, R, F, W, S, D, P\}$. ■

The next lemma is a useful tool for the proofs of Lemmas 6.3.8, 6.3.9, and 6.3.10.

Lemma 6.3.7 *Let* $X_i \in \{\underleftarrow{W}, \underleftarrow{S}\}$ *or* $X_i \in \{E_R, R, F, \underleftarrow{D}\}$ *with* $\mathit{FIRST}(\mathtt{flip}_i, \mathtt{left})$. *Furthermore, let* $X_{i+1} \in \{\underrightarrow{W}, \underrightarrow{S}\}$ *or* $X_{i+1} \in \{E_R, R, F, \underrightarrow{D}\}$ *with* $\mathit{FIRST}(\mathtt{flip}_{i+1}, \mathtt{right})$. *Then the first of the two processes* i *or* $i+1$ *testing its second resource enters* P *after having performed this test (if this time ever comes).*

Proof. By Lemma 6.3.4 Res_i is free. Moreover, Res_i is the second resource needed by both i and $i+1$. Whichever tests for it first gets it and enters P. ■

Lemma 6.3.8 *If* $X_i = \underleftarrow{S}$ *and* $X_{i+1} \in \{\underrightarrow{W}, \underrightarrow{S}\}$ *then, eventually, one of the two processes* i *or* $i+1$ *enters* P. *The same result holds if* $X_i \in \{\underleftarrow{W}, \underleftarrow{S}\}$ *and* $X_{i+1} = \underrightarrow{S}$.

Proof. Being in state S, process i tests its second resource eventually. An application of Lemma 6.3.7 finishes the proof. ■

Lemma 6.3.9 *Let* $X_i = \underleftarrow{S}$ *and* $X_{i+1} \in \{E_R, R, F, \underrightarrow{D}\}$. *If* $\mathit{FIRST}(\texttt{flip}_{i+1}, \texttt{right})$, *then, eventually, one of the two processes* i *or* $i+1$ *enters* P. *The same result holds if* $X_i \in \{E_R, R, F, D\}$, $X_{i+1} = \underrightarrow{S}$ *and* $\mathit{FIRST}(\texttt{flip}_i, \texttt{left})$.

Proof. Being in state S, process i tests its second resource eventually. An application of Lemma 6.3.7 finishes the proof. ■

Lemma 6.3.10 *Assume that* $X_{i-1} \in \{E_R, R, T\}$, $X_i = \underleftarrow{W}$, *and* $X_{i+1} \in \{E_R, R, F, \underrightarrow{W}, \underrightarrow{D}\}$. *If* $\mathit{FIRST}(\texttt{flip}_{i-1}, \texttt{left})$ *and* $\mathit{FIRST}(\texttt{flip}_{i+1}, \texttt{right})$, *then eventually one of the three processes* $i-1$, i *or* $i+1$ *enters* P.

Proof. Let s be a state of M such that $X_{i-1}(s) \in \{E_R, R, T\}$, $X_i(s) = \underleftarrow{W}$, and $X_{i+1}(s) \in \{E_R, R, F, \underrightarrow{W}, \underrightarrow{D}\}$. Let $\mathcal{A}$ be an adversary of *Fairadvs*, and let α be an extended execution of $\Omega_{prexec(M,\{s\},\mathcal{A})}$ where the result of the first coin flip of process $i-1$ is `left` and the result of the first coin flip of process $i+1$ is `right`. By Lemma 6.3.6, eventually either process $i-1$ reaches configuration P in α or process i reaches configuration $\underleftarrow{S}$ in α. If $i-1$ reaches configuration P, then we are done. If not, then let $\alpha = \alpha_1 \frown \alpha_2$ such that $\mathit{lstate}(\alpha_1)$ is the first state s' of α with $X_i(s') = \underleftarrow{S}$. If $i+1$ enters P before the end of α_1, then we are done. Otherwise, $X_{i+1}(\mathit{fstate}(\alpha_2))$ is either in $\{\underrightarrow{W}, \underrightarrow{S}\}$ or it is in $\{E_R, R, F, \underrightarrow{D}\}$ and process $i+1$ has not flipped any coin yet in α. From the finite-history-insensitivity of *Fairadvs* we can then apply Lemma 6.3.7: eventually process i tests its second resource and by Lemma 6.3.7 process i enters P if process $i+1$ did not check its second resource in the meantime. If process $i+1$ checks its second resource before process i does the same, then by Lemma 6.3.7 process $i+1$ enters P. ■

Lemma 6.3.11 *Assume that* $X_{i+2} \in \{E_R, R, T\}$, $X_{i+1} = \underrightarrow{W}$, *and* $X_i \in \{E_R, R, F, \underleftarrow{W}, \underleftarrow{D}\}$. *If* $\mathit{FIRST}(\texttt{flip}_i, \texttt{left})$ *and* $\mathit{FIRST}(\texttt{flip}_{i+2}, \texttt{right})$, *then eventually one of the three processes* i, $i+1$ *or* $i+2$, *enters* P.

Proof. The proof is analogous to the one of Lemma 6.3.10. This lemma is the symmetric case of Lemma 6.3.10. ■

Proposition 6.3.12 *Starting from a global configuration in* $\mathcal{G}$, *then, with probability at least* $1/4$, *some process enters* P *eventually. Equivalently:*

$$\mathcal{G} \xrightarrow[1/4]{} \mathcal{P}.$$

Proof. Lemmas 6.3.8 and 6.3.9 jointly treat the case where $X_i = \underleftarrow{S}$ and $X_{i+1} \in \{E_R, R, F, \underrightarrow{\#}\}$ and the symmetric case where $X_i \in \{E_R, R, F, \underleftarrow{\#}\}$ and $X_{i+1} = \underrightarrow{S}$; Lemmas 6.3.10 and 6.3.11 jointly treat the case where $X_i = \underleftarrow{W}$ and $X_{i+1} \in \{E_R, R, F, \underrightarrow{W}, \underrightarrow{D}\}$ and the symmetric case where $X_i \in \{E_R, R, F, \underleftarrow{W}, \underleftarrow{D}\}$ and $X_{i+1} = \underrightarrow{W}$.

Specifically, each lemma shows that a compound event of the kind $\mathit{FIRST}(\texttt{flip}_i, x)$ and $\mathit{FIRST}(\texttt{flip}_j, y)$ leads to $\mathcal{P}$. Each of the basic events $\mathit{FIRST}(\texttt{flip}_i, x)$ has probability at least $1/2$. From Lemma 6.2.4 each of the compound events has probability at least $1/4$. Thus the probability of reaching $\mathcal{P}$ eventually is at least $1/4$. ■

We now turn to $\mathcal{F} \xrightarrow[1/2]{} \mathcal{G} \cup \mathcal{P}$. The proof is divided in two parts and constitute the global argument of the proof of progress, i.e., the argument that focuses on the whole system rather than on a couple of processes.

Lemma 6.3.13 *Start with a state s of $\mathcal{F}$. If there exists a process i for which $X_i(s) = F$ and $(X_{i-1}, X_{i+1}) \neq (\underset{\rightarrow}{\#}, \underset{\leftarrow}{\#})$, then, with probability at least $1/2$ a state of $\mathcal{G} \cup \mathcal{P}$ is reached eventually.*

Proof. If $s \in \mathcal{G} \cup \mathcal{P}$, then the result is trivial. Let s be a state of $\mathcal{F} - (\mathcal{G} \cup \mathcal{P})$ and let i be such that $X_i(s) = F$ and $(X_{i-1}, X_{i+1}) \neq (\underset{\rightarrow}{\#}, \underset{\leftarrow}{\#})$. Assume without loss of generality that $X_{i+1} \neq \underset{\leftarrow}{\#}$, i.e., $X_{i+1} \in \{E_R, R, F, \underset{\rightarrow}{\#}\}$. The case for $X_{i-1} \neq \underset{\rightarrow}{\#}$ is similar. Furthermore, we can assume that $X_{i+1} \in \{E_R, R, F, \underset{\rightarrow}{D}\}$ since if $X_{i+1} \in \{\underset{\rightarrow}{W}, \underset{\rightarrow}{S}\}$ then s is already in $\mathcal{G}$. We show that the event schema $\mathit{FIRST}((\texttt{flip}_i, \texttt{left}), (\texttt{flip}_{i+1}, \texttt{right}))$, which by Lemma 6.2.2 has probability at least $1/2$, leads eventually to a state of $\mathcal{G} \cup \mathcal{P}$. Let $\mathcal{A}$ be an adversary of *Fairadvs*, and let α be an extended execution of $\Omega_{prexec(M,\{s\},\mathcal{A})}$ where if process i flips before process $i+1$ then process i flips left, and if process $i+1$ flips before process i then process $i+1$ flips right.

Then, eventually, i performs one transition and reaches W. Let $j \in \{i, i+1\}$ be the first of i and $i+1$ that reaches W and let s_1 be the state reached after the first time process j reaches W. If some process reached P in the meantime, then we are done. Otherwise there are two cases to consider. If $j = i$, then, $\texttt{flip}_i$ yields $\texttt{left}$ and $X_i(s_1) = \underset{\leftarrow}{W}$ whereas X_{i+1} is (still) in $\{E_R, R, F, \underset{\rightarrow}{D}\}$. Therefore, $s_1 \in \mathcal{G}$. If $j = i+1$, then $\texttt{flip}_{i+1}$ yields $\texttt{right}$ and $X_{i+1}(s_1) = \underset{\rightarrow}{W}$ whereas $X_i(s_1)$ is (still) F. Therefore, $s_1 \in \mathcal{G}$. ■

Lemma 6.3.14 *Start with a state s of $\mathcal{F}$. If there exists a process i for which $X_i(s) = F$ and $(X_{i-1}(s), X_{i+1}(s)) = (\underset{\rightarrow}{\#}, \underset{\leftarrow}{\#})$. Then, with probability at least $1/2$, a state of $\mathcal{G} \cup \mathcal{P}$ is reached eventually.*

Proof. The hypothesis can be summarized into the form $(X_{i-1}(s), X_i(s), X_{i+1}(s)) = (\underset{\rightarrow}{\#}, F, \underset{\leftarrow}{\#})$. Since $i-1$ and $i+1$ point in different directions, by moving to the right of $i+1$ there is a process k pointing to the left such that process $k+1$ either points to the right or is in $\{E_R, R, F, P\}$, i.e., $X_k(s) \in \{\underset{\leftarrow}{W}, \underset{\leftarrow}{S}, \underset{\leftarrow}{D}\}$ and $X_{k+1}(s) \in \{E_R, R, F, \underset{\rightarrow}{W}, \underset{\rightarrow}{S}, \underset{\rightarrow}{D}, P\}$.

If $X_k(s) \in \{\underset{\leftarrow}{W}, \underset{\leftarrow}{S}\}$ and $X_{k+1}(s) \neq P$ then $s \in \mathcal{G}$ and we are done; if $X_{k+1}(s) = P$ then $s \in \mathcal{P}$ and we are done. Thus, we can restrict our attention to the case where $X_k(s) = \underset{\leftarrow}{D}$.

We show that $\mathit{FIRST}((\texttt{flip}_k, \texttt{left}), (\texttt{flip}_{k+1}, \texttt{right}))$, which by Lemma 6.2.2 has probability at least $1/2$, leads eventually to $\mathcal{G} \cup \mathcal{P}$. Let $\mathcal{A}$ be an adversary of *Fairadvs*, and let α be an extended execution of $\Omega_{prexec(M,\{s\},\mathcal{A})}$ where if process k flips before process $k+1$ then process k flips left, and if process $k+1$ flips before process k then process $k+1$ flips right.

Then, eventually, process k performs at least two transitions and hence goes to configuration W. Let $j \in \{k, k+1\}$ be the first of k and $k+1$ that reaches W and let s_1 be the state reached after the first time process j reaches W. If some process reached P in the meantime, then we are done. Otherwise, we distinguish two cases. If $j = k$, then, $\mathtt{flip}_k$ yields `left` and $X_k(s_1) = \underleftarrow{W}$ whereas X_{k+1} is (still) in $\{E_R, R, F, \underrightarrow{\#}\}$. Therefore, $s_1 \in \mathcal{G}$. If $j = k + 1$, then $\mathtt{flip}_{k+1}$ yields `right` and $X_{k+1}(s_1) = \underrightarrow{W}$ whereas $X_k(s_1)$ is (still) in $\{\underleftarrow{D}, F\}$. Therefore, $s_1 \in \mathcal{G}$. ■

Proposition 6.3.15 *Start with a state s of $\mathcal{F}$. Then, with probability at least* $1/2$, *a state of $\mathcal{G} \cup \mathcal{P}$ is reached eventually. Equivalently:*

$$\mathcal{F} \xrightarrow[1/2]{} \mathcal{G} \cup \mathcal{P}.$$

Proof. The hypothesis of Lemmas 6.3.13 and 6.3.14 form a partition of $\mathcal{F}$. ■

Finally, we prove $\mathcal{RT} \xrightarrow[1]{} \mathcal{F} \cup \mathcal{G} \cup \mathcal{P}$.

Proposition 6.3.16 *Starting from a state s of $\mathcal{RT}$, then a state of $\mathcal{F} \cup \mathcal{G} \cup \mathcal{P}$ is reached eventually. Equivalently:*

$$\mathcal{RT} \xrightarrow[1]{} \mathcal{F} \cup \mathcal{G} \cup \mathcal{P}.$$

Proof. Let s be a state of $\mathcal{RT}$. If $s \in \mathcal{F} \cup \mathcal{G} \cup \mathcal{P}$, then we are trivially done. Suppose that $s \notin \mathcal{F} \cup \mathcal{G} \cup \mathcal{P}$. Then in s each process is in $\{E_R, R, W, S, D\}$ and there exists at least process in $\{W, S, D\}$. Let $\mathcal{A}$ be an adversary of *Fairadvs*, and let α be an extended execution of $\Omega_{prexec(M,\{s\},\mathcal{A})}$.

We first argue that eventually some process reaches a state of $\{S, D, F\}$ in α. This is trivially true if in state s there is some process in $\{S, D\}$. If this is not the case, then all processes are either in E_R or R or W. Eventually, some process in R or W performs a transition. If the first process not in E_R performing a transition started in E_R or R, then it reaches F and we are done; if the first process performing a transition is in W, then it reaches S since in s no resource is held. Once a process i is in $\{S, D, F\}$, then eventually process i reaches either state F or P, and we are done. ■

6.4 General Coin Lemmas

The coin lemmas of Section 6.2 are sufficiently general to prove the correctness of the Randomized Dining Philosophers algorithm of Lehmann and Rabin. However, there are several other coin events that are relevant for the analysis of distributed algorithms. For example, the toy resource allocation protocol that we used in Chapter 5 cannot be verified yet. In this section we present two general coin lemmas: the first one deals with multiple outcomes in a random draw; the second one gives a generalization of all the coin lemmas presented in the thesis. Unfortunately, generality and simplicity are usually incompatible: the two coin lemmas of this section are conceptually more complicated than those of Section 6.2.

6.4.1 Conjunction of Separate Coin Events with Multiple Outcomes

The coin lemma of Section 6.2.4 deals with the result of the intersection of several coin events. Thus, for example, if each coin event expresses the process of flipping a coin, then the coin lemma of Section 6.2.4 can be used to study the probability that all the coins yield head.

However, we may be interested in the probability that at least half of the coins yield head, or in the probability that exactly 5 coins yield head. The coin lemmas of Section 6.2 are not adequate. Suppose now that we use each coin event to express the process of rolling a dice. The coin events of Section 6.2 are not adequate again since they can deal only with binary outcomes: we can observe only whether a specific set U is reached or not. How can we express the event that for each number i between 1 and 6 there is at least one dice that rolls i?

In this section we define a coin event and prove a coin lemma that can deal with the scenarios outlined above. Let M be a probabilistic automaton, and let $\mathcal{S}$ be a set of n tuples $\{x_1, \ldots, x_n\}$, where for each i, $1 \leq i \leq n$, x_i is a tuple $(a_i, U_{i,1}, \ldots, U_{i,k})$ consisting of an action of M and k pairwise disjoint sets of states of M. Let the actions a_i be all distinct. Let E be a set of tuples $((1, j_1), \ldots, (n, j_n))$ where for each i, $1 \leq i \leq n$, the value of j_i is between 1 and k. For each extended execution α of M and each i, $1 \leq i \leq n$, let

$$U_i(\alpha) = \begin{cases} \{(i,1), \ldots, (i,k)\} & \text{if } a_i \text{ does not occur} \\ \{(i,j)\} & \text{if } a_i \text{ occurs and its first occurrence leads to } U_{i,j} \\ \emptyset & \text{otherwise.} \end{cases}$$

Then define $GFIRST(\mathcal{S}, E)$ to be the function that associates with each probabilistic execution fragment H of M the set of extended executions α of Ω_H such that $E \cap (U_1(\alpha \triangleright q_0^H) \times \cdots \times U_k(\alpha \triangleright q_0^H)) \neq \emptyset$.

We illustrate the definition above by encoding the dice rolling example. In each tuple $(a_i, U_{i,1}, \ldots, U_{i,k})$ a_i identifies the action of rolling the i^{th} dice, $k = 6$, and for each j, $U_{i,j}$ is the set of states where the i^{th} dice rolls j. The set E identifies the set of outcomes that are considered to be good. In the case of the dices E is the set of tuples $((1, j_1), \ldots, (n, j_n))$ where for each number l between 1 and 6 there is at least one i such that $j_i = l$. The function $U_i(\alpha)$ checks whether the i^{th} dice is rolled and identifies the outcome. If the dice is not rolled, then, we allow any outcome as a possible one; if the dice is rolled and hits $U_{i,j}$, then the outcome is (i, j); if the the dice is rolled and the outcome is not in any one of the sets $U_{i,j}$'s, then there is no outcome (this case does not arise in our example). Then, an extended execution α of Ω_H is in the event $GFIRST(\mathcal{S}, E)(H)$ if at least one of the outcomes associated with $\alpha \triangleright q_0^H$ is an element of E, i.e., if by choosing the outcome of the dices that are not rolled in $\alpha \triangleright q_0^H$ all the six numbers appear as the outcome of some dice.

Let p be the probability that by rolling n dices all the six numbers appear as the outcome of some dice. Then, the lemma below states that $P_H[GFIRST(\mathcal{S}, E)(H)] \geq p$ for each H.

Proposition 6.4.1 *Let M be a probabilistic automaton. Let $\mathcal{S}$ be a set of n tuples $\{x_1, \ldots, x_n\}$ where for each i, $1 \leq i \leq n$, x_i is a tuple $(a_i, U_{i,1}, \ldots, U_{i,k})$ consisting of an action of M and k pairwise disjoint sets of states of M. Let the actions a_i be all distinct. Let E be a set of tuples $((1, j_1), \ldots, (n, j_n))$ where for each i, $1 \leq i \leq n$, the value of j_i is between 1 and k. For each i, j, $1 \leq i \leq n$, $1 \leq j \leq k$, let $p_{i,j}$ be a real number between 0 and 1 such that for each transition $(s, \mathcal{P})$ of M where $P[a_i] > 0$, $P[U_{i,j}|a_i] \geq p_{i,j}$, and let $\mathcal{C}$ be the collection of the $p_{i,j}$s. Let $P_{\mathcal{C}}[E]$ be the probability of the event E assuming that each experiment i is run independently, and that for each i a pair (i, j) is chosen with probability $p_{i,j}$. Then, for each probabilistic execution fragment H of M, $P_H[GFIRST(\mathcal{S}, E)(H)] \geq P_{\mathcal{C}}[E]$.*

Proof. For each state q of H, each $i \in \{1, \ldots, n\}$, and each $j \in \{1, \ldots, k\}$, denote by $\Omega(q, U_{i,j})$ the set $\{(a_i, q') \in \Omega_q^H \mid lstate(q') \in U_{i,j}\}$ of pairs where a_i occurs and leads to a state of $U_{i,j}$, and denote by $\Omega(q,\overline{U_i})$ the set $\{(a_i,q') \in \Omega_q^H \mid \mathit{lstate}(q') \notin \cup_j U_{i,j}\}$ of pairs where a_i occurs and none of the $U_{i,j}$s is reached. For each $i \in \{1,\dots,n\}$, let Θ_i be the set of states q of H such that no action a_j, $1 \leq j \leq n$, occurs in $q \triangleright q_0^H$, and $P_q^H[a_i] > 0$.

We prove the lemma by induction on n. If $n = 1$ then the result follows from Lemma 6.2.1 (the event can be transformed into a new event with two outcomes); otherwise,

$$P_H[\overline{\mathit{GFIRST}(\mathcal{S},E)(H)}] = \sum_{i\in\{1,\dots,n\}}\sum_{q\in\Theta_i} P_H[C_q]\left(\left(\sum_{(a_i,q')\in\Omega(q,\overline{U_i})} P_q^H[(a_i,q')]\right) + \left(\sum_{j\in\{1,\dots,k\}}\sum_{(a_i,q')\in\Omega(q,U_{i,j})} P_q^H[(a_i,q')]P_{H\triangleright q'}[\overline{\mathit{GFIRST}(\mathcal{S}_i,E_{(i,j)})(H\triangleright q')}]\right)\right). \tag{6.23}$$

where $\mathcal{S}_i$ is obtained from $\mathcal{S}$ by removing the tuple $(a_i, U_{i,1},\dots,U_{i,k})$, and $E_{(i,j)}$ is the set of tuples $((1,j_1),\dots,(i-1,j_{i-1}),(i+1,j_{i+1}),\dots,(n,j_n))$ such that $((1,j_1),\dots,(i-1,j_{i-1}),(i,j),(i+1,j_{i+1}),\dots,(n,j_n)) \in E$. Let $\mathcal{C}_i$ be obtained from $\mathcal{C}$ by removing all the probabilities of the form $p_{i,j}$, $1 \leq j \leq k$. Then, by induction,

$$P_{H\triangleright q'}[\overline{\mathit{GFIRST}(\mathcal{S}_i,E_{(i,j)})(H\triangleright q')}] \leq (1 - P_{\mathcal{C}_i}[E_{(i,j)}]). \tag{6.24}$$

From the properties of conditional probabilities and the definition of $\mathcal{C}$,

$$P_{\mathcal{C}_i}[E_{(i,j)}] = P_{\mathcal{C}}[E|(i,j)]. \tag{6.25}$$

Thus, by using (6.24) and (6.25) in (6.23), and by expressing $P_q^H[(a_i,q')]$ as $P_q^H[a_i]P_q^H[(a_i,q')|a_i]$, we obtain

$$P_H[\overline{\mathit{GFIRST}(\mathcal{S},E)(H)}] \leq \sum_{i\in\{1,\dots,n\}}\sum_{q\in\Theta_i} P_H[C_q]P_q^H[a_i]\left(\left(\sum_{(a_i,q')\in\Omega(q,\overline{U_i})} P_q^H[(a_i,q')|a_i]\right) + \left(\sum_{j\in\{1,\dots,k\}}\sum_{(a_i,q')\in\Omega(q,U_{i,j})} P_q^H[(a_i,q')|a_i](1 - P_{\mathcal{C}}[E|(i,j)])\right)\right). \tag{6.26}$$

For each i, j and q, let $p_{i,j,q}$ be $P_q^H[\Omega(q,U_{i,j})|a_i]$. Then, from (6.26),

$$P_H[\overline{\mathit{GFIRST}(\mathcal{S},E)(H)}] \leq \sum_{i\in\{1,\dots,n\}}\sum_{q\in\Theta_i} P_H[C_q]P_q^H[a_i] \left((1 - p_{i,1,q} - \cdots - p_{i,k,q}) + \left(\sum_{j\in\{1,\dots,k\}} p_{i,j,q}(1 - P_{\mathcal{C}}[E|(i,j)])\right)\right), \tag{6.27}$$

which becomes

$$P_H[\overline{\mathit{GFIRST}(\mathcal{S},E)(H)}] \leq \sum_{i\in\{1,\dots,n\}}\sum_{q\in\Theta_i} P_H[C_q]P_q^H[a_i]\left(1 - \sum_{j\in\{1,\dots,k\}} P_{\mathcal{C}}[E|(i,j)]p_{i,j,q}\right) \tag{6.28}$$

after some simple algebraic simplifications. Using the same argument as in the proof of Lemma 6.2.1, for each i, j and each q, $p_{i,j,q} \geq p_{i,j}$. Thus,

$$P_H[\overline{GFIRST(\mathcal{S},E)(H)}] \leq \sum_{i\in\{1,\ldots,n\}} \sum_{q\in\Theta_i} P_H[C_q] P_q^H[a_i] \left(1 - \sum_{j\in\{1,\ldots,k\}} P_{\mathcal{C}}[E|(i,j)] p_{i,j}\right). \tag{6.29}$$

Finally, observe that $\sum_{i\in\{1,\ldots,n\}} \sum_{q\in\Theta_i} P_H[C_q] P_q^H[a_i]$ is the probability that some action a_i occurs, and observe that $\sum_{j\in\{1,\ldots,k\}} P_{\mathcal{C}}[E|(i,j)] p_{i,j} = P_{\mathcal{C}}[E]$. Thus,

$$P_H[\overline{GFIRST(\mathcal{S},E)(H)}] \leq 1 - P_{\mathcal{C}}[E] \tag{6.30}$$

■

6.4.2 A Generalized Coin Lemma

All the coin lemmas that we have studied in this chapter share a common characteristic. Given a probabilistic execution fragment H, we identify n separate classes of random draws to observe. Each class can be observed at most once in every execution α of Ω_H, and if any class cannot be observed, then we allow for any arbitrary outcome. In this section we formalize this idea.

Let H be a probabilistic execution fragment of a probabilistic automaton M. A *coin-event specification* for H is a collection C of tuples $(q, X, X_1, \ldots, X_k)$ consisting of a state of H, a subset X of Ω_q^H, and m pairwise disjoint subsets of X, such that the following properties are satisfied:

1. for each state q of H there is at most one tuple of C whose state is q;
2. for each state q of H such that there exists a tuple of C with state q, there is no prefix q' of q such that there exists a tuple $(q', X, X_1, \ldots, X_k)$ in C and a pair (a, q'') in X where q'' is a prefix of q.

The set C is the object that identifies one of the classes of random draws to be observed. For each transition tr_q^H and each tuple $(q, X, X_1, \ldots, X_k)$ of C, the set X identifies the part of tr_q^H that is relevant for C, and the sets $X_1, \ldots, X_k$ identify some of the possible outcomes. The first requirement for C guarantees that there is at most one way to observe what happens from a state q of H, and the second requirement states that along every execution of Ω_H there is at most one place where C is observed.

As an example, consider the observation of whether the first occurrence of an action a, which represents a coin flip, leads to head. Then C is the set of tuples (q, X, X_1) where action a does not occur in $q \triangleright q_0^H$ and $P_q^H[a] > 0$, X is the set of pairs of Ω_q^H where action a occurs, and X_1 is the set of pairs of X where the coin flips head.

Let α be an extended execution of Ω_H, and let q be a state of H such that $q \leq \alpha$. We say that C *occurs* in α at q iff there exists a tuple $(q, X, X_1, \ldots, X_k)$ in C and a pair (a, q') in X such that $q' \leq \alpha$. Moreover, if $(a, q') \in X_j$, we say that C occurs in α at q and leads to X_j.

Two coin event specifications C_1 and C_2 are said to be *separate* iff from every state q of H, if $(q, X_1, X_{1,1}, \ldots, X_{1,k})$ is a tuple of C_1 and $(q, X_2, X_{2,1}, \ldots, X_{2,k})$ is a tuple of C_2, then $X_1 \cap X_2 = \emptyset$. In other words, there is no interference between the observations of C_1 and the observations of C_2. Let $\mathcal{S} = \{C_1, \ldots, C_n\}$ be a set of pairwise *separate* coin-event specifications. For notational convenience, for each $i \in \{1, \ldots, n\}$ and each state q of H such that there exists a tuple in C_i with state q, denote such tuple by $(q, X_{q,i}, X_{q,i,1}, \ldots, X_{q,i,k})$

Let E be a set of tuples $((1, j_1), \ldots, (n, j_n))$ where for each i, $1 \leq i \leq n$, the value of j_i is between 1 and k. For each extended execution α of Ω_H and each i, $1 \leq i \leq n$, let

$$U_i(\alpha) = \begin{cases} \{(i,1), \ldots, (i,k)\} & \text{if } C_i \text{ does not occur in } \alpha \\ \{(i,j)\} & \text{if } C_i \text{ occurs in } \alpha \text{ leading to } X_{q,i,j} \\ \emptyset & \text{otherwise.} \end{cases}$$

Then, define $GCOIN(\mathcal{S}, E)(H)$ to be the set of extended executions of Ω_H such that $E \cap (U_1(\alpha \triangleright q_0^H) \times \cdots \times U_k(\alpha \triangleright q_0^H)) \neq \emptyset$.

Lemma 6.4.2 *Let H be a probabilistic execution fragment of a probabilistic automaton M. Let $\mathcal{S} = \{C_1, \ldots, C_n\}$ be a set of separate coin-event specifications for H. For each i, j, $1 \leq i \leq n$, $1 \leq j \leq k$, let $p_{i,j}$ be a real number between* 0 *and* 1 *such that for each $i \in \{1, \ldots, n\}$ and each tuple $(q, X_{q,i}, X_{q,i,1}, \ldots, X_{q,i,m})$ of C_i, $P_q^H[X_{q,i,j} | X_{q,i}] \geq p_{i,j}$. Let $\mathcal{C}$ be the collection of the $p_{i,j}$'s. Let $P_{\mathcal{C}}[E]$ be the probability of the event E assuming that each experiment i is run independently, and for each i a pair (i,j) is chosen with probability $p_{i,j}$. Then, $P_H[GCOIN(\mathcal{S}, E)(H)] \geq P_{\mathcal{C}}[E]$.*

Proof. For each state q of H and each i, $1 \leq i \leq n$, if there exists a tuple in C_i with state q, then denote $X_{q,i} \backslash \cup_{j \in \{1,\ldots,k\}} X_{q,i,j}$ by $\overline{X_{q,i}}$. For each i, $1 \leq i \leq n$, let Θ_i be the set of states q of H such that there exists a tuple with state q in C_i and no coin-event C_j, $1 \leq j \leq n$, occurs in $q \triangleright q_0^H$.

We prove the lemma by induction on n, using $n = 0$ for the base case. For $n = 0$ we assume that $P[E] = 1$ and that $GCOIN(\mathcal{S}, E)(H) = \Omega_H$. In this case the result is trivial. Otherwise,

$$P_H[\overline{GCOIN(\mathcal{S}, E)(H)}] = \sum_{i \in \{1,\ldots,n\}} \sum_{q \in \Theta_i} P_H[C_q] \left(\left(\sum_{(a,q') \in \overline{X_{q,i}}} P_q^H[(a,q')] \right) \right.$$
$$\left. + \left(\sum_{j \in \{1,\ldots,k\}} \sum_{(a,q') \in X_{q,i,j}} P_q^H[(a,q')] P_{H \triangleright q'}[\overline{GCOIN(\mathcal{S} \triangleright q', E_{(i,j)})(H \triangleright q')}] \right) \right). \tag{6.31}$$

where $\mathcal{S} \triangleright q'$ is obtained from $\mathcal{S}$ by removing C_i and, for each $j \neq i$, by transforming the set C_j into $\{(q \triangleright q', X \triangleright q', X_1 \triangleright q', \ldots, X_k \triangleright q') \mid (q, X, X_1, \ldots, X_k) \in C_j, q' \leq q\}$. Then, by induction,

$$P_{H \triangleright q'}[\overline{GCOIN(\mathcal{S} \triangleright q', E_{(i,j)})(H \triangleright q')}] \leq (1 - P_{\mathcal{C}_i}[E_{(i,j)}]). \tag{6.32}$$

From the properties of conditional probabilities and the definition of $\mathcal{C}$,

$$P_{\mathcal{C}_i}[E_{(i,j)}] = P_{\mathcal{C}}[E | (i,j)]. \tag{6.33}$$

Thus, by using (6.32) and (6.33) in (6.31), and expressing $P_q^H[(a,q')]$ as $P_q^H[X_{q,i}] P_q^H[(a,q') | X_{q,i}]$, we obtain

$$P_H[\overline{GCOIN(\mathcal{S}, E)(H)}] \leq \sum_{i \in \{1,\ldots,n\}} \sum_{q \in \Theta_i} P_H[C_q] P_q^H[X_{q,i}] \left(\left(\sum_{(a,q') \in \overline{X_{q,i}}} P_q^H[(a,q') | X_{q,i}] \right) \right.$$
$$\left. + \left(\sum_{j \in \{1,\ldots,k\}} \sum_{(a,q') \in X_{q,i,j}} P_q^H[(a,q') | X_{q,i}] (1 - P_{\mathcal{C}}[E | (i,j)]) \right) \right). \tag{6.34}$$

For each i, j and q, let $p_{i,j,q}$ be $P_q^H[X_{q,i,j}|X_{q,i}]$. Then, from (6.34),

$$P_H[\overline{GCOIN(\mathcal{S}, E)(H)}] \leq \sum_{i \in \{1,\ldots,n\}} \sum_{q \in \Theta_i} P_H[C_q] P_q^H[X_{q,i}]$$
$$\left((1 - p_{i,1,q} - \cdots - p_{i,k,q}) + \left(\sum_{j \in \{1,\ldots,k\}} p_{i,j,q}(1 - P_{\mathcal{C}}[E|(i,j)]) \right) \right), \tag{6.35}$$

which becomes

$$P_H[\overline{GCOIN(\mathcal{S}, E)(H)}]$$
$$\leq \sum_{i \in \{1,\ldots,n\}} \sum_{q \in \Theta_i} P_H[C_q] P_q^H[X_{i,j}] \left(1 - \sum_{j \in \{1,\ldots,k\}} P_{\mathcal{C}}[E|(i,j)] p_{i,j,q} \right) \tag{6.36}$$

after some simple algebraic simplifications. From hypothesis, for each i, j and each q, $p_{i,j,q} \geq p_{i,j}$. Thus,

$$P_H[\overline{GCOIN(\mathcal{S}, E)(H)}]$$
$$\leq \sum_{i \in \{1,\ldots,n\}} \sum_{q \in \Theta_i} P_H[C_q] P_q^H[X_{q,i}] \left(1 - \sum_{j \in \{1,\ldots,k\}} P_{\mathcal{C}}[E|(i,j)] p_{i,j} \right). \tag{6.37}$$

Finally, observe that $\sum_{i \in \{1,\ldots,n\}} \sum_{q \in \Theta_i} P_H[C_q] P_q^H[X_{q,i}]$ is the probability that some C_i occurs, and observe that $\sum_{j \in \{1,\ldots,k\}} P_{\mathcal{C}}[E|(i,j)] p_{i,j} = P_{\mathcal{C}}[E]$. Thus,

$$P_H[\overline{GCOIN(\mathcal{S}, E)(H)}] \leq 1 - P_{\mathcal{C}}[E] \tag{6.38}$$

■

6.5 Example: Randomized Agreement with Stopping Faults

In this section we analyze the Randomized Agreement algorithm of Ben-Or [BO83]. Its proof of correctness is an application of Lemma 6.4.2. The proof that we present in this section is not as detailed as the proof of the Dining Philosophers algorithm, but contains all the information necessary to fill in all the details, which we leave to the reader.

6.5.1 The Problem

Consider n asynchronous processes that communicate through a network of reliable channels (i.e., channels that deliver all the messages in the same order as they are received, and that never fail to deliver a message), and suppose that each process i starts with an initial value $v_i \in \{0, 1\}$. Suppose that each process can broadcast a message to every other process in a single operation. Each process runs an algorithm that at some point may decide on one value of $\{0, 1\}$. Each process decides at most once. The algorithm should be designed so that the following properties are satisfied.

1. **Agreement:** all the processes that decide choose the same value.

2. **Validity:** if all the processes have the same initial value v, then v is the only possible decision value.

3. **f-failure termination:** if at most f processes fail, then all the non-failing processes decide a value.

We assume that a process fails by stopping, i.e., by failing to send messages to other processes from some point on. Since the processes are asynchronous, no processes can distinguish a slow process from a failing process.

Unfortunately, it is known from [FLP85] that there is no deterministic algorithm for asynchronous processes that solves the agreement problem and guarantees 1-failure termination. Here we present the randomized algorithm of Ben-Or [BO83], which solves the agreement problem with certainty, and guarantees f-failure termination with probability 1 whenever $n > 3f$.

6.5.2 The Algorithm

Each process i has local variables x, initially v_i, and y, initially $null$, and executes a series of *stages* numbered $1, 2, \ldots$, each stage consisting of two *rounds*. Each process runs forever, even after it decides. At stage $st \geq 1$, process i does the following.

1. Broadcast (first, st, v), where v is the current value of x, and then wait to obtain $n - f$ messages of the form $(\mathit{first}, st, *)$, where $*$ stands for any value. If all the messages have the same value v, then set $y := v$, otherwise set $y := null$.

2. Broadcast $(second, st, v)$, where v is the current value of y, and then wait to obtain $n - f$ messages of the form $(second, st, *)$. There are three cases:

   (a) if all the messages have the same value $v \neq null$, then set $x := v$ and perform a $decide(v)_i$ operation if no decision was made already;

   (b) if at least $n - 2f$ messages, but not all the messages, have the same value $v \neq null$, then set $x := v$ without deciding (the assumption $n > 3f$ guarantees that there cannot be two different such values v);

   (c) otherwise, set x to 0 with probability 1/2 and to 1 with probability 1/2.

The intuition behind the use of randomness is that at each stage, if a decision is not made yet, with probability at least $1/2^n$ all the processes that choose a value at random choose the same "good" value. Thus, with probability 1 there is eventually a stage where the processes that choose a value at random choose the same good value, and this leads to a decision.

We now give an idea of the structure of the probabilistic automaton M that describes Ben-Or's algorithm. Each process i has the two variables x and y mentioned in the description of the algorithm, plus a queue m_j for each process j that records the unprocessed messages received from process j, initially $null$, a stage counter st, initially 1, a program counter pc, and a boolean variable $decided$ that is set to $true$ iff process i has decided already. There is a channel $C_{i,j}$ between every pair of processes. Each channel $C_{i,j}$ is essentially a buffer like the buffer described in Chapter 3 (cf. Figure 3-1), whose inputs are actions of the form $(\mathit{first}, st, v)_i$ and $(second, st, v)_i$, and whose outputs are actions of the form $(\mathit{first}, st, v)_{i,j}$ and $(second, st, v)_{i,j}$. To broadcast a message (first, st, v), process i performs the action $(\mathit{first}, st, v)_i$.

A message $(\mathit{first}, \mathit{st}, v)$ is received by process i from process j through the action $(\mathit{first}, \mathit{st}, v)_{j,i}$. The definition of the transition relation of M is straightforward.

6.5.3 The High Level Proof

Agreement and validity are easy to prove and do not involve any probabilistic argument.

Lemma 6.5.1 *Ben-Or's algorithm satisfies the agreement and validity conditions.*

Proof. We start with validity. Suppose that all the processes start with the same value v. Then it is easy to see that every process that completes stage 1 decides on v in that stage. This is because the only value sent or received by any process in the first round is v, and thus the only value sent or received by any process in the second round is v, leading to the decision of v.

For agreement, suppose that some process decides, and let process i be the first process that decides. Let v and st be the value decided by process i and the stage at which process i decides, respectively. Then it must be the case that process i receives $n - f$ $(\mathit{second}, \mathit{st}, v)$ messages. This implies that any other process j that completes stage st receives at least $n - 2f$ $(\mathit{second}, \mathit{st}, v)$ messages, since it hears from all but at most f of the processes that process i hears from. This means that process j cannot decide on a value different from v at stage st; moreover, process j sets $x := v$ at stage st. Since this is true for all the processes that complete stage st, then an argument similar to the argument for validity shows that any process that completes stage $\mathit{st} + 1$ and does not decide in stage st decides v at stage $\mathit{st} + 1$. ■

The argument for f-failure termination involves probability. We assume that all the processes but at most f are scheduled infinitely many times. Thus, let *f-fair* be the set of adversaries for M such that for each probabilistic execution fragment H generated by an adversary of *f-fair* the set Ω_H contains only executions of M where at least $n - f$ processes are scheduled infinitely many times. It is easy to check that *f-fair* is finite-history-insensitive.

Let $\mathcal{B}$ be the set of reachable states of M; let $\mathcal{F}$ be the set of reachable states of M where no process has decided yet and there exists a value st and a number i such that process i received exactly $n - f$ messages $(\mathit{first}, \mathit{st}, *)$, and no other process has ever received more than $n - f - 1$ messages $(\mathit{first}, \mathit{st}, *)$; finally, let $\mathcal{O}$ be the set of reachable states of M where at least one process has decided.

It is easy to show that

$$\mathcal{B} \xrightarrow[1]{}_{\mathit{f\text{-}fair}} \mathcal{F} \cup \mathcal{O}. \tag{6.39}$$

Specifically, let α be an f-fair execution fragment of M starting from a reachable state s of M, and let st be the maximum value of the stages reached by each process in s. Then, stage $\mathit{st} + 1$ is reached eventually in α, and thus there is a state s' in α where some process is the first one to receive $n - f$ messages $(\mathit{first}, \mathit{st} + 1, *)$. The state s' is a state of $\mathcal{F} \cup \mathcal{O}$.

In Section 6.5.4 we show that

$$\mathcal{F} \xrightarrow[1/2^n]{} \mathcal{O}. \tag{6.40}$$

Thus, combining (6.39) and (6.40) with Theorem 5.5.2, and by using Proposition 5.5.6, we obtain

$$\mathcal{B} \xrightarrow[1]{} \mathcal{O}. \tag{6.41}$$

Finally, we need to show that in every f-fair execution where at least one process decides all the non-failing processes decide eventually. This is shown already in the second part of the proof of Lemma 6.5.1.

6.5.4 The Low Level Proof

In this section we prove the progress statement of (6.40) using the generalized coin lemma. Consider a state s of $\mathcal{F}$, and let i be the process that has received $n-f$ messages (first, st, v). Let $\mathcal{A}$ be an adversary of *f-fair*, and let H be $\mathit{prexec}(M, \mathcal{A}, s)$.

For each j, $1 \leq j \leq n$, let C_j be the set of triplets (q, X, X_1) where q is a state of H such that process j is at stage st in $\mathit{lstate}(q)$ and there is a non-zero probability that process j chooses randomly between 0 and 1 from q, X is the set of pairs of Ω_q^H where process j performs a transition, and X_1 is defined as follows. Let s' be $\mathit{lstate}(q)$, and let v be a *good* value if at least $f+1$ of the messages $(\mathit{first}, st, *)$ processed by process i have value v. We emphasize the word "processed" since, although each process can receive more that $n-f$ messages $(\mathit{first}, st, *)$, only $n-f$ of those messages are used (processed).

1. If 0 is a good value, then let X_1 be the set of pairs of X where process i chooses 0;

2. if 1 is a good value and 0 is not a good value, then let X_1 be the set of pairs of X where process i chooses 1.

Observe that in s' there is at least one good value, and at most two values; thus, C_j is well defined. It is easy to check that $C_1, \ldots, C_n$ are separate coin event specifications; moreover, for each j, $1 \leq j \leq n$, and each triplet (q, X, X_1) of C_j, $P_q^H[X_1|X] = 1/2$. Let $E = \{((1,1),(2,1),\ldots,(n,1)\}$. From Lemma 6.4.2, $P_H[\mathit{GCOIN}((C_1,\ldots,C_n),E)(H)] \geq 1/2^n$.

We are left with the proof that in each extended execution of $\mathit{GCOIN}((C_1,\ldots,C_n),E)(H)$ all the non-faulty processes choose a value. More precisely, we show that the non-faulty processes complete stage st setting x to the same value v. Then, the second part of the proof of Lemma 6.5.1 can be used to show that all the non-faulty processes decide on v at the end of stage $st+1$; in particular at least one process decides. We distinguish two cases.

1. In s' there is exactly one good value v.

   In this case every other process receives at least one copy of v during the first round of stage st, and thus y is set either to v or to *null*. Therefore, v is the only value that a process chooses by a non-random assignment at the end of stage st. On the other hand, if a process j chooses a value at random at the end of stage st, the definition of C_j guarantees that the value chosen is v. Thus, every process that completes stage st sets $x := v$.

2. In s' there are two good values.

   In this case every process receives at least one copy of 0 and one copy of 1, and thus y is set to *null*. Therefore, each process chooses a value at random at the end of stage st. The definition of $C_1, \ldots, C_n$ guarantees that every process that completes stage st sets $x := 0$.

6.6 Example: The Toy Resource Allocation Protocol

Lemma 6.4.2 can be used also to prove formally that the toy resource allocation protocol of Section 5.1 guarantees that, under any deterministic fair oblivious adversary (cf. Example 5.6.2 for the definition of a fair oblivious adversary), process M_1 eventually gets a resource. This result can be extended to general oblivious adversaries by using the results about deterministic and randomized adversaries proved in Chapter 5 (cf. Proposition 5.7.11).

Recall from Example 6.1.1 that we want to identify a coin event that expresses the following property: the first coin flip of M_1 after the first coin flip of M_2 is different from the last coin flip of M_2 before the first time M_1 checks its resource after flipping. In the rest of the section we specify two coin event specifications C_1 and C_2. The specification C_1 identifies the first coin flip of M_1 after the first coin flip of M_2, while the specification C_2 identifies the last coin flip of M_2 before the first time M_1 checks its resource after flipping.

Let H be a probabilistic execution fragment, generated by a deterministic fair oblivious adversary, such that the first state of q_0^H is reachable in M. Let C_1 be the set of tuples (q, X, X_1, X_2) where

1. q is a state of H such that M_2 flips at least once in $q \triangleright q_0^H$, M_1 does not flip in $q \triangleright q_0^H$ after the first time M_2 flips, and M_1 flips from q,
2. X is the set Ω_q^H,
3. X_1 is the set of pairs of X where M_1 flips head,
4. X_2 is the set of pairs of X where M_1 flips tail.

Observe that C_1 is a coin-event specification. Moreover, observe that for each tuple of C_1, $P_q^H[X_1|X] = 1/2$ and $P_q^H[X_2|X] = 1/2$. Let C_2 be the set of tuples (q, X, X_1, X_2) where

1. q is a state of H such that either
   (a) M_1 does not flip in $q \triangleright q_0^H$ after M_2 flips, M_2 flips from q, and there exists a state $q' \geq q$ such that M_2 flips exactly once in $q' \triangleright q$ and M_1 flips and checks its resource after flipping in $q' \triangleright q$, or
   (b) M_1 flips and does not check its resource after the first flip of M_2 in $q \triangleright q_0^H$, M_2 flips from q, and there exists a state $q' \geq q$ such that M_2 flips exactly once in $q' \triangleright q$, M_1 does not check its resource in $q' \triangleright q$, and M_1 checks its resource from q',
2. X is the set Ω_q^H,
3. X_1 is the set of pairs of X where M_2 flips head,
4. X_1 is the set of pairs of X where M_2 flips tail.

Informally, C_2 identifies the coin flip of M_2 that precedes the point where M_1 checks the resource determined by C_1. Figure 6-4 illustrates graphically the two cases of the definition of C_2. Observe that for each tuple of C_2, $P_q^H[X_1|X] = 1/2$ and $P_q^H[X_2|X] = 1/2$. Since H is generated by an oblivious deterministic adversary, then it is easy to verify that C_2 is a coin-event specification. The important point is to verify that Condition 2 of the definition of a coin event is satisfied; this is the point where the fact that an adversary is oblivious and deterministic is used.

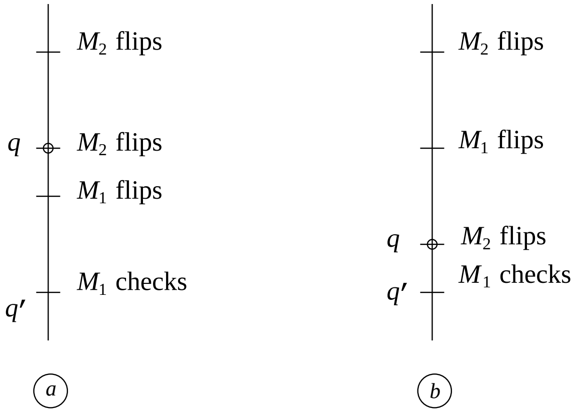


Figure 6-4: The definition of C_2 for the toy resource allocation protocol.

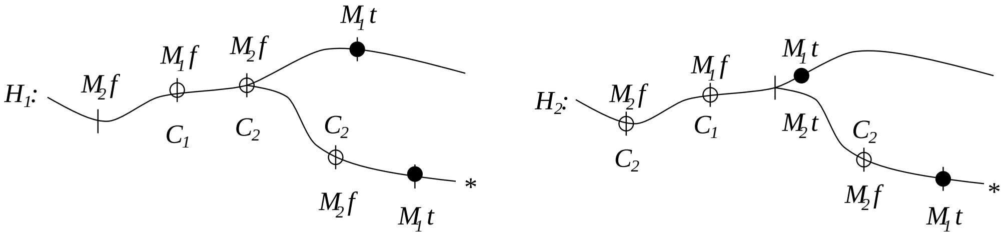


Figure 6-5: How C_2 could not be a coin event specification.

Example 6.6.1 (How C_2 could not be a coin event specification.) To give a rough idea of why Condition 2 does not fail, Figure 6-5 shows how Condition 2 could fail. Consider the execution of H_1 that is marked with $*$, and denote it by α; denote by α' the other execution of H_1 that appears in the figure. The unfilled circles mark the points where a coin event specification is observed. By following α from left to right we observe C_1 and then we observe C_2. The reason why we observe C_2 the first time is that along α' M_1 tests its resource. However, continuing to follow α, we observe C_2 again because along α M_2 tests its resource later. Using oblivious adversaries we are guaranteed that such a situation does not arise because if along α' M_1 tests its resource before M_2 flips again, then the same property holds along α.

The probabilistic execution H_2 of Figure 6-5 illustrates how Condition 2 can fail by using randomized schedulers. After M_1 flips, the adversary chooses randomly whether to let M_1 test its resource (higher filled circle) or to let M_2 continue. ■

Let E be the set $\{((1,1)(2,2)),((1,2),(2,1))\}$, which expresses the fact that C_1 and C_2 yield two different outcomes. It is easy to check that in every execution of $GCOIN((C_1,C_2),E)(H)$ M_1 eventually gets one resource. Thus, from Lemma 6.4.2, the probability that M_1 gets its resource in H is at least $1/4$. Since H is a generic probabilistic execution fragment, then, under any deterministic fair oblivious adversary M_1 gets a resource eventually with probability at least $1/4$. Since the set of deterministic fair oblivious adversaries is finite-history-insensitive, Lemma 5.5.6 applies, and we conclude that under any deterministic fair oblivious adversary M_1 gets a resource eventually with probability 1.

6.7 The Partition Technique

Even though the coin lemmas can be used to prove the correctness of several nontrivial algorithms, two of which have been illustrated in this chapter, there are algorithms for which the coin lemmas do not seem to be suitable. One example of such an algorithm is the randomized algorithm for maximal independent sets of Awerbuch, Cowen and Smith [ACS94]; another example is the toy resource allocation protocol again.

Example 6.7.1 (The coin lemmas do not work always) In Section 6.6 we have shown that the toy resource allocation protocol guarantees progress against fair oblivious adversaries; however, in Example 5.6.2 we have stated that the toy resource allocation protocol guarantees progress also against adversaries that do not know only the outcome of those coins that have not been used yet. Such a result cannot be proved using the coin lemmas of this chapter because situations like those outlined in Example 6.6.1 arise. For example, after the first time M_2 flips, we could schedule M_2 again and then schedule M_1 to test its resource only if M_2 gets the resource R_1.

Another way to obtain a situation where the coin lemmas of this chapter do not apply is to modify the second instruction of the resource allocation protocol as follows

2. if the chosen resource is free, then get it, *otherwise go back to 1*. ■

Example 6.7.1 shows us that some other techniques need to be developed; it is very likely that several new techniques will be discovered by analyzing other algorithms. In this section we hint at a proof technique that departs considerably from the coin lemmas and that is sufficiently powerful to deal with the toy resource allocation protocol. We illustrate the technique with an example.

Example 6.7.2 (The partition technique) Let $\mathcal{A}$ be a generic fair adversary for the toy resource allocation protocol that does not know the outcome of those coin flips that have not been used yet, and let H be a probabilistic execution generated by $\mathcal{A}$. Assume for simplicity that $\mathcal{A}$ is deterministic; the result for a generic adversary follows from Proposition 5.7.11. Consider an element of Ω_H, and consider the first point q where M_1 flips a coin (cf. Figure 6-6). The coin flipping transition leads to two states q_h and q_t that are not distinguishable by $\mathcal{A}$, which means that from q_h and q_t the adversary schedules the same process. If the process scheduled from q_h and q_t is M_2, then the states reached from q_h are in one-to-one correspondence with the states reached from q_t, since they differ only in the value of the coin flipped by M_1. Figure 6-6 illustrates the case where M_2 flips a coin. Furthermore, two corresponding states are reached with the same probability. The one-to-one correspondence between the states reached form q_h and q_t is maintained until M_1 tests its chosen resource.

Consider now a point where M_1 tests its resource. Figure 6-6 illustrates four of these points, denoted by $q_{t,1}$, $q_{h,1}$, $q_{t,2}$, and $q_{h,2}$. If M_1 fails to obtain the resource, it means that M_2 holds that resource at that point. However, M_2 holds the same resource in the corresponding state via the one-to-one correspondence M_2, while M_1 tests the other resource. Thus, M_1 succeeds in getting the chosen resource. (cf. states $q_{t,1}$ and $q_{h,1}$ of Figure 6-6.

The bottom line is that we have partitioned the states where M_1 checks its resource in two sets, and we have shown that for each pair of corresponding states there is at least one state where M_1 succeeds in getting a resource. In some cases, like for states $q_{t,2}$, and $q_{h,2}$ of

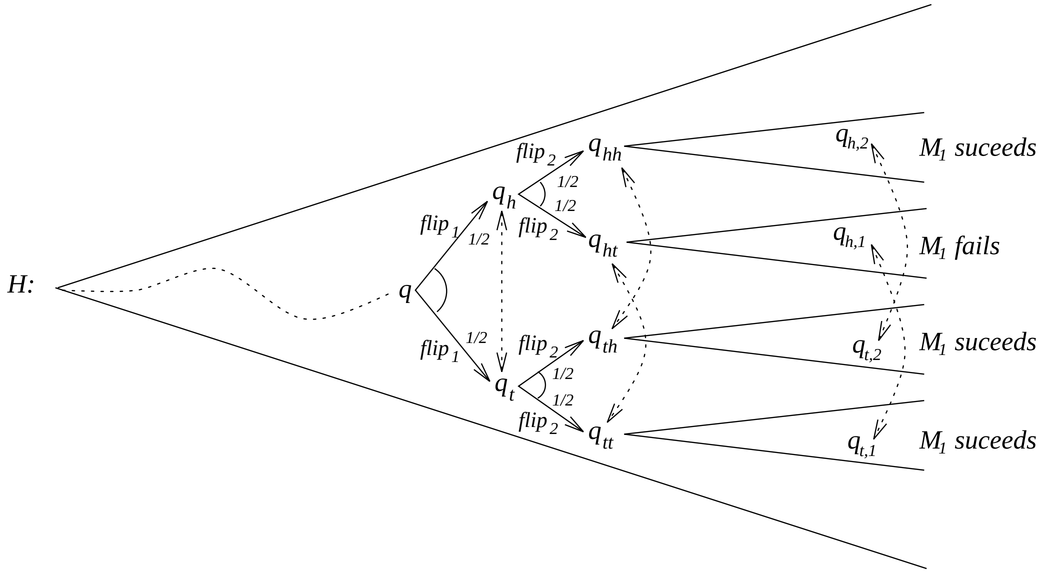


Figure 6-6: The partition technique.

Figure 6-6, M_1 succeeds in getting its resource from both the corresponding states (M_2 does not hold any resource). Thus, M_1 gets a resource with probability at least 1/2. ■

6.8 Discussion

To our knowledge, no techniques similar to our coin lemmas or to our partition technique were proposed before; however, similar arguments appear in several informal analysis of randomized algorithms. The idea of reducing the analysis of a randomized algorithm to the analysis of an ordinary pure nondeterministic system was at the base of the qualitative analysis techniques described in Sections 2.5.1 and 2.5.2. Here we have been able to apply the same idea for a quantitative analysis of an algorithm.

In this chapter we have focused mainly on how to apply a coin lemma for the verification of a randomized algorithm; once a good coin event is identified, the analysis is reduced to verify properties of a system that does not contain randomization. We have carried out this last part using detailed operational arguments, which can be error prone themselves. However, since the problem is reduced to the analysis of a non-randomized system, several existing techniques can be used to eliminate our operational arguments. In [PS95] Segala and Pogosyants show how such an analysis can be carried out formally and possibly mechanized.

Chapter 7

Hierarchical Verification: Trace Distributions

7.1 Introduction

So far we have defined a model to describe randomized concurrent and distributed systems, and we have shown how to study the properties of a system by means of a direct analysis of its structure. A specification is a set of properties that an implementation should satisfy, and an implementation is a probabilistic automaton that satisfies the desired properties.

Another approach to the analysis of a system considers an automaton as a specification itself. Then, an abstract notion of *observation* is defined on automata, and an automaton is said to be an implementation of another automaton iff there is a specific relation, usually a preorder relation, between their abstract observations. Examples of observations are traces [Hoa85, LV91] (cf. Section 3.2.3), and failures [Hoa85, BHR84]; in these two cases implementation is expressed by set inclusion.

7.1.1 Observational Semantics

Formally, an automaton A is associated with a set $Obs(A)$ of observations, and a preorder relation $\mathcal{R}$ is defined over sets of observations (for example $\mathcal{R}$ can be set inclusion). Then, an automaton A_1 is said to implement another automaton A_2, denoted by $A_1 \sqsubseteq A_2$, iff $Obs(A_1)\ \mathcal{R}\ Obs(A_2)$. The function $Obs()$ is called an *observational semantics*, or alternatively a *behavioral semantics*; in the second case the observations are thought as the possible behaviors of an automaton.

The methodology based on preorder relations is an instance of the hierarchical verification method: a specification, which is usually very abstract, can be refined successively into less abstract specifications, each one implementing the more abstract specification, till the actual implementation is obtained. Figure 7-1 gives an example of a specification that is refined two times to build the actual implementation. Of course it is implicitly assumed that the relevant properties of a system are only those that are preserved by the chosen implementation relation. Thus, given a relation, it is important to understand what properties it preserves. Coarse relations may not preserve all the relevant properties, but they are usually easy to verify, i.e., it is usually easy to establish whether such a relation holds; finer relations that preserve exactly the

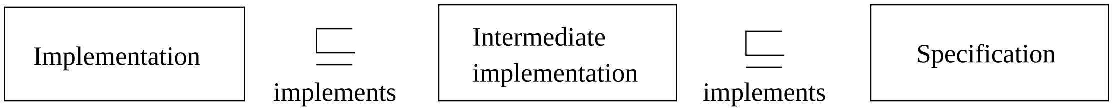


Figure 7-1: Refinement of a specification.

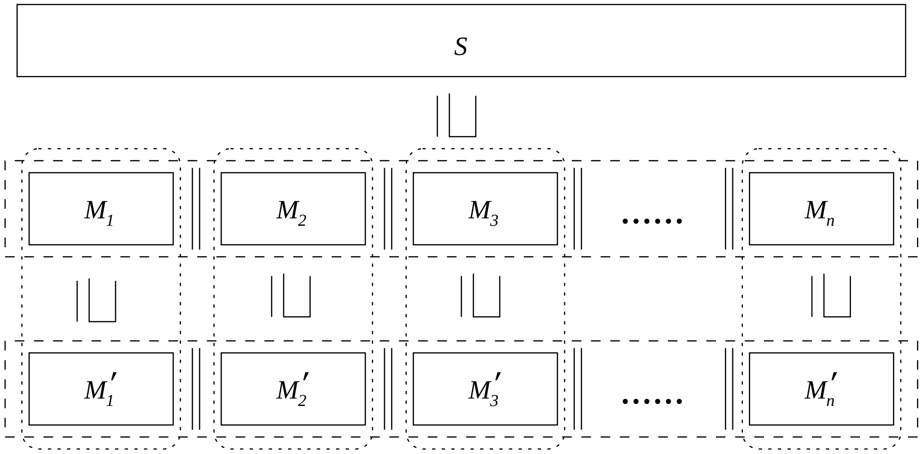


Figure 7-2: Modular design.

relevant properties are usually difficult to characterize and verify; other relations that preserve all the relevant properties and that are easy to verify are usually too fine, i.e., they distinguish too much. Some tradeoff is necessary.

7.1.2 Substitutivity and Compositionality

When the size of a problem becomes large, it is common to decompose the problem into simpler subproblems that are solved separately. Figure 7-2 gives an example. A large specification S is decomposed into several subcomponents $M_1, \ldots, M_n$ that interact together to implement S. For example, a complex computer system can be described by the interaction of a central processor unit, a memory unit, and an Input/Output unit. Then, each subcomponent specification M_i is given to a development team that builds an implementation M_i'. Finally, the implementations are put together to build an actual implementation of S. This kind of approach is called *modular design*; however, in order to guarantee the soundness of modular design, we need to guarantee that an implementation works properly in every context where its specification works properly, i.e., our implementation relation must be preserved by parallel composition (i.e., it must be a *precongruence*). This property is called *substitutivity* of a preorder relation, and constitutes one of the most important properties that an implementation relation should satisfy.

A property that is strictly related to the substitutivity of $\sqsubseteq$ is called *compositionality* of *Obs*(). That is, there is an operator $\|$ defined on pairs of sets of observations such that $Obs(A_1 \| A_2) = Obs(A_1) \| Obs(A_2)$. Compositionality and substitutivity are used interchangeably when talking informally about concurrent systems, and it is easy to get confused by the meanings of the two terms. To clarify every doubt, here is how the two concepts are related.

Theorem 7.1.1 *Let Obs() be an observational semantics, $\mathcal{R}$ be an equivalence relation over sets of observations, and let, for each set x of observations, $[x]_{\mathcal{R}}$ be the equivalence class of x under $\mathcal{R}$. Let $A_1 \equiv A_2$ iff $Obs(A_1)\ \mathcal{R}\ Obs(A_2)$. Then the following two statements are equivalent.*

1. *$\equiv$ is substitutive, i.e., if $A_1 \equiv A_2$ then for each A_3, $A_1 \| A_3 \equiv A_2 \| A_3$;*

2. *Obs() is compositional, i.e., there exists an operator $\|$ on equivalence classes of observations such that $[Obs(A_1 \| A_2)]_{\mathcal{R}} = [Obs(A_1)]_{\mathcal{R}} \| [Obs(A_1)]_{\mathcal{R}}$.* ■

If $\mathcal{R}$ is set equality, then we can remove the equivalence classes from the second statement since each set of observations is an equivalence class. The substitutivity of a preorder relation is stronger than the substitutivity of its kernel equivalence relation, since the direction of the inequality must be preserved under parallel composition. For this reason our primary concern in this chapter is the substitutivity of the implementation relation.

7.1.3 The Objective of this Chapter

In this chapter we study the simplest implementation relation based on observations, i.e., trace inclusion, and we extend the corresponding precongruence to the probabilistic framework. The trace preorder constitutes the basis for several other implementation relations and is known to preserve the *safety* properties of a system [AS85]. Roughly speaking, a safety property says that "something good holds forever" or that "something bad does not happen". The trace preorder is important for ordinary automata for its simplicity and for the availability of the *simulation method* [LT87, Jon91, LV91] (cf. Chapter 8), which provides several sufficient conditions for the trace preorder relation to hold. Other relations, based either on failures [Hoa85, BHR84] or on any other form of enriched traces, can be obtained by following the same methodology that we present here.

In the probabilistic framework a trace is replaced by a *trace distribution*, where the trace distribution of a probabilistic execution fragment H is the distribution over traces induced by $\mathcal{P}_H$, the probability space associated with H. The trace distribution preorder is defined as inclusion of trace distributions.

Unfortunately, the trace distribution preorder is not a precongruence (cf. Example 7.4.1), which in turn means that the observational semantics based on trace distributions is not compositional. A standard approach in this case is to define the *trace distribution precongruence* as the coarsest precongruence that is contained in the trace distribution preorder; then, in order to have a compositional observational semantics that captures the trace distribution precongruence, an alternative, more operational and constructive characterization of the trace distribution precongruence is derived. We give an alternative characterization of the trace distribution precongruence by exhibiting a context, called the *principal context*, that distinguishes two probabilistic automata whenever there exists a distinguishing context. This leads to the notion of a *principal trace distribution*, which is a trace distribution of a probabilistic automaton in parallel with the principal context; the trace distribution precongruence can be characterized alternatively as inclusion of principal trace distributions.

Several other characterizations of the trace distribution precongruence could be found, possibly leading to different observational semantics equivalent to the principal trace distribution semantics. Further experience with each one of the alternative semantics will determine which

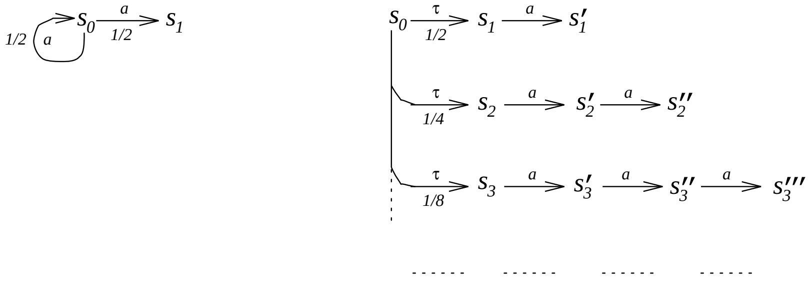


Figure 7-3: Trace distribution equivalent probabilistic automata.

one is more useful. One of the problems with the principal trace distribution characterization is that, although from Theorem 7.1.1 there exists an operator $\|$ defined on principal traces, the definition of $\|$ is not simple. For ordinary automata the traces of a parallel composition of two automata are exactly those sequences of actions that restricted to each component give a trace of the component. This property does not hold for principal trace distributions (cf. Example 7.4.1). It is desirable to find a semantics that characterizes the trace distribution precongruence and for which the corresponding parallel composition operator has a simple definition; however, it is not clear whether such a semantics exists.

7.2 Trace Distributions

Let H be a probabilistic execution fragment of a probabilistic automaton M, and let f be a function from Ω_H to $\Omega = ext(H)^* \cup ext(H)^\omega$ that assigns to each execution of Ω_H its trace. The *trace distribution* of H, denoted by $tdistr(H)$, is the probability space $completion((\Omega, \mathcal{F}, P))$ where $\mathcal{F}$ is the σ-field generated by the cones C_β, where β is a finite trace of H, and $P = f(P_H)$. Observe that, from Proposition 3.1.4, f is a measurable function from $(\Omega_H, \mathcal{F}_H)$ to $(\Omega, \mathcal{F})$, since the inverse image of a cone is a union of cones. Denote a generic trace distribution by $\mathcal{D}$. A trace distribution of a probabilistic automaton M is the trace distribution of one of the probabilistic executions of M. Denote by $tdistrs(M)$ the set of the trace distributions of a probabilistic automaton M.

It is easy to see that trace distributions extend the traces of ordinary automata: the trace distribution of a linear probabilistic execution fragment α is a distribution that assigns probability 1 to $trace(\alpha)$.

Given two probabilistic execution fragments H_1 and H_2, it is possible to check whether $tdistr(H_1) = tdistr(H_2)$ just by verifying that $P_{tdistr(H_1)}[C_\beta] = P_{tdistr(H_2)}[C_\beta]$ for each finite sequence of actions β. This is an easy consequence of the extension theorem (cf. Theorem 3.1.2).

Example 7.2.1 (Reason for the definition of Ω) The reader may wonder why we have not defined Ω to be $trace(\Omega_H)$. This is to avoid to distinguish two trace distribution just because they have different sample spaces. Figure 7-3 illustrates the idea. The two probabilistic automata of Figure 7-3 have the same trace distributions; however, the left probabilistic automaton has a probabilistic execution where the trace a^∞ occurs with probability 0, while the right probabilistic automaton does not. Thus, by defining the sample space of $tdistr(H)$ to be $trace(\Omega_H)$, the two probabilistic automata of Figure 7-3 would be distinct. In Chapter 8 we define several simulation relations for probabilistic automata, and we show that they are sound for the trace distribution precongruence; such a result would not be true with the alternative definition of a trace distribution. ■

Prefixes

The notion of a prefix for traces can be extended to the probabilistic framework by following the same idea as for the notion of a prefix defined on probabilistic executions (cf. Section 4.2.6). A trace distribution $\mathcal{D}$ is a *prefix* of a trace distribution $\mathcal{D}'$, denoted by $\mathcal{D} \leq \mathcal{D}'$, iff for each finite trace β, $P_{\mathcal{D}}[C_\beta] \leq P_{\mathcal{D}'}[C_\beta]$. Thus, two trace distributions are equal iff each one is a prefix of the other.

Lemma 7.2.1 *Let H_1 and H_2 be two probabilistic execution fragments of a probabilistic automaton M. If $H_1 \leq H_2$, then $tdistr(H_1) \leq tdistr(H_2)$.* ■

Action Restriction

Similarly to the ordinary case, it is possible to define an action restriction operator on trace distributions. Let $\mathcal{D} = (\Omega, \mathcal{F}, P)$ be a trace distribution, and let V be a set of actions. Then the *restriction* of $\mathcal{D}$ to V, denoted by $\mathcal{D} \upharpoonright V$, is the probability space *completion*$((\Omega', \mathcal{F}', P'))$ where $\Omega' = \Omega \upharpoonright V$, $\mathcal{F}'$ is the σ-field generated by the sets of cones of Ω', and P' is the inverse image of P under the function that restricts traces to V.

Lemma 7.2.2 *Let $\mathcal{D}$ be a trace distribution. Then $(\mathcal{D} \upharpoonright V_1) \upharpoonright V_2 = \mathcal{D} \upharpoonright (V_1 \cap V_2)$.*

Proof. This is a direct consequence of the fact that restricting a trace to V_1 and then to V_2 is equivalent to restricting the same trace to $V_1 \cap V_2$. Formally, $\cdot \upharpoonright (V_1 \cap V_2) = (\cdot \upharpoonright V_2) \circ (\cdot \upharpoonright V_1)$. ■

Finally, we want to show that, if $M = M_1 \| M_2$, then the projection of a trace distribution of M onto M_1 and M_2 is a trace distribution of M_1 and M_2, respectively. Formally,

Proposition 7.2.3 *If $\mathcal{D} \in tdistrs(M_1 \| M_2)$, then $\mathcal{D} \upharpoonright acts(M_i) \in tdistrs(M_i)$, $i = 1, 2$.*

The converse of Proposition 7.2.3 is not true; an illustrating example is given in Section 7.4 (cf. Example 7.4.1). The rest of this section is dedicated to the proof of Proposition 7.2.3. We start with a definition of an *internal trace distribution*, which is a trace distribution that does not abstract from internal actions.

Let α be an execution of a probabilistic automaton M. The *internal trace* of α, denoted by *itrace*(α), is the subsequence of α consisting of the actions of M. Let H be a probabilistic execution fragment of M, and let f be a function from Ω_H to $\Omega = acts(H)^* \cup acts(H)^\omega$ that assigns to each execution of Ω_H its internal trace. The *internal trace distribution* of H, denoted by *itdistr*(H), is the probability space *completion*$((\Omega, \mathcal{F}, P))$ where $\mathcal{F}$ is the σ-field generated by the cones of Ω, and $P = f(P_H)$. Observe that, from Proposition 3.1.4, f is a measurable function from $(\Omega_H, \mathcal{F}_H)$ to $(\Omega, \mathcal{F})$. Denote a generic internal trace distribution by $\mathcal{D}$. Denote the set of internal trace distributions of a probabilistic automaton M by *itdistrs*(M).

Lemma 7.2.4 *Let H be a probabilistic execution fragment of a probabilistic automaton M. Then, $tdistr(H) = itdistr(H) \upharpoonright ext(H)$.*

Proof. This is a direct consequence of the fact that the set of executions of H whose trace contains a given β is the set of executions of H whose internal trace restricted to the external actions of H contains β. Formally, $\mathit{trace}(\cdot) = \mathit{itrace}(\cdot) \circ (\cdot \restriction \mathit{ext}(H))$. ■

Lemma 7.2.5 *Let H be a probabilistic execution fragment of $M_1 \| M_2$, where M_1 and M_2 are two compatible probabilistic automata. Then $\mathit{itdistr}(H\lceil M_i) = \mathit{itdistr}(H) \restriction \mathit{acts}(M_i)$, $i = 1, 2$.*

Proof. Let $\mathcal{P}$ denote $\mathit{itdistr}(H\lceil M_i)$, and let $\mathcal{P}'$ denote $\mathit{itdistr}(H) \restriction \mathit{acts}(M_i)$. We need to show that for each finite internal trace β, $P[C_\beta] = P'[C_\beta]$. Let $\mathcal{P}''$ denote $\mathit{itdistr}(H)$. From the definition of an internal trace,

$$P[C_\beta] = P_{H\lceil M_i}[\alpha \in \Omega_{H\lceil M_i} \mid \beta \leq \mathit{itrace}(\alpha)]. \tag{7.1}$$

From the definition of $\mathcal{P}'$ and $\mathcal{P}''$,

$$P'[C_\beta] = P''[\beta' \in \Omega'' \mid \beta \leq \beta' \restriction \mathit{acts}(M_i)]. \tag{7.2}$$

From the definition of $\mathit{itdistr}(H)$ and (7.2),

$$P'[C_\beta] = P_H[\alpha \in \Omega_H \mid \beta \leq \mathit{itrace}(\alpha) \restriction \mathit{acts}(M_i)]. \tag{7.3}$$

Thus, from (7.1) and (7.3), we need to show that

$$P_{H\lceil M_i}[\alpha \in \Omega_{H\lceil M_i} \mid \beta \leq \mathit{itrace}(\alpha)] = P_H[\alpha \in \Omega_H \mid \beta \leq \mathit{itrace}(\alpha) \restriction \mathit{acts}(M_i)]. \tag{7.4}$$

By using a characterization of the involved events as a disjoint union of cones, and by rewriting Equation 7.4 accordingly, we obtain

$$\begin{aligned} & P_{H\lceil M_i}[\bigcup_{q \in \mathit{states}(H\lceil M_i) \mid \mathit{itrace}(q)=\beta, \mathit{lact}(q)=\mathit{lact}(\beta)} C_q] \qquad (7.5)\\ = \; & P_H[\bigcup_{q \in \mathit{states}(H) \mid \mathit{itrace}(q) \restriction \mathit{acts}(M_i)=\beta, \mathit{lact}(q)=\mathit{lact}(\beta)} C_q]. \end{aligned}$$

Observe that for each $q \in \mathit{states}(H)$ such that $\mathit{itrace}(q) \restriction \mathit{acts}(M_i) = \beta$ and $\mathit{lact}(q) = \mathit{lact}(\beta)$, the state $q\lceil M_i$ is a state of $H\lceil M_i$ such that $\mathit{itrace}(q\lceil M_i) = \beta$ and $\mathit{lact}(q\lceil M_i) = \mathit{lact}(\beta)$. Moreover, the states q of the left expression of (7.5) are partitioned by the relation that relates q and q' whenever $q\lceil M_i = q'\lceil M_i$. Thus, if we show that for each trace β and each $q \in \mathit{states}(H\lceil M_i)$ such that $\mathit{itrace}(q) = \beta$ and $\mathit{lact}(q) = \mathit{lact}(\beta)$,

$$P_{H\lceil M_i}[C_q] = P_H[\cup_{q' \in q\rceil H \mid \mathit{lact}(q')=\mathit{lact}(\beta)} C_{q'}], \tag{7.6}$$

Equation (7.5) is proved. Observe that

$$P_H[\cup_{q' \in \mathit{states}(H) \mid q'\lceil M_i = q, \mathit{lact}(q')=\mathit{lact}(\beta)} C_{q'}] = \sum_{q' \in \mathit{min}(q\rceil H)} P_H[C_{q'}], \tag{7.7}$$

since $\{q' \in \mathit{states}(H) \mid q'\lceil M_i = q, \mathit{lact}(q') = \mathit{lact}(\beta)\} = \mathit{min}(q\rceil H)$. Thus, Equation (7.6) becomes

$$P_{H\lceil M_i}[C_q] = \sum_{q' \in \mathit{min}(q\rceil H)} P_H[C_{q'}], \tag{7.8}$$

which is true from Proposition 4.3.5. ■

Lemma 7.2.6 *Let H be a probabilistic execution fragment of $M_1 \| M_2$, where M_1 and M_2 are two compatible probabilistic automata. Then $tdistr(H\lceil M_i) = tdistr(H) \upharpoonright acts(M_i)$.*

Proof. From Lemma 7.2.4,

$$tdistr(H\lceil M_i) = itdistr(H\lceil M_i) \upharpoonright ext(M_i). \tag{7.9}$$

From Lemma 7.2.5 and (7.9),

$$tdistr(H\lceil M_i) = (itdistr(H) \upharpoonright acts(M_i)) \upharpoonright ext(M_i). \tag{7.10}$$

From Lemma 7.2.2 and (7.10),

$$tdistr(H\lceil M_i) = (itdistr(H) \upharpoonright ext(H)) \upharpoonright acts(M_i). \tag{7.11}$$

From Lemma 7.2.4 and (7.11),

$$tdistr(H\lceil M_i) = tdistr(H) \upharpoonright acts(M_i), \tag{7.12}$$

which is what we needed to prove. ■

Proof of Proposition 7.2.3. Let $\mathcal{D} \in tdistrs(M_1 \| M_2)$. Then there exists a probabilistic execution H of $M_1 \| M_2$ such that $tdistr(H) = \mathcal{D}$. From Proposition 4.3.4, $H\lceil M_i$ is a probabilistic execution of M_i. From Lemma 7.2.6, $tdistr(H\lceil M_i) = \mathcal{D} \upharpoonright acts(M_i)$. Thus, $\mathcal{D} \upharpoonright acts(M_i) \in tdistrs(M_i)$. ■

7.3 Trace Distribution Preorder

Once trace distributions are defined, the trace distribution preorder can be defined as trace distribution inclusion. Formally, let M_1, M_2 be two probabilistic automata with the same external action signature. The *trace distribution preorder* is defined as follows.

$$M_1 \sqsubseteq_D M_2 \text{ iff } tdistrs(M_1) \subseteq tdistrs(M_2). \tag{7.13}$$

The trace distribution preorder is a conservative extension of the trace preorder of ordinary automata, and it preserves properties that resemble the safety properties of ordinary automata [AS85]. Here we give some examples of such properties.

Example 7.3.1 The following property is preserved by the trace distribution preorder.

> *"After some finite trace β has occurred, then the probability that some other trace β' occurs, is not greater than p."*

In fact, suppose that $M_1 \sqsubseteq_D M_2$, and suppose that M_2 satisfies the property above, while M_1 does not. Then there is a trace distribution of M_1 where the probability of β' after β conditional to β is greater than p. Since $M_1 \sqsubseteq_D M_2$, there is a trace distribution of M_2 where the probability of β' after β conditional to β is greater than p. This contradicts the hypothesis that M_2 satisfies the property above. Observe that the property above would still be preserved if we replace β' with a set of traces. ■

Example 7.3.2 The following property is preserved by the trace distribution preorder.

> "*In every computation where infinite external activity occurs with probability* 1*, if a finite trace β occurs, then the probability that some other trace β' occurs after β given that β occurs is at least p.*"

A more concrete instantiation of the property above is "under the hypothesis that a distributed system never deadlocks, every request of service eventually gets a response with probability at least p". This property is definitely more interesting than the property of Example 7.3.1 since it involves a progress statement, one of the property of key interest for the analysis of randomized distributed algorithms. Thus, if in a system it is always possible to avoid a deadlock, under the assumption that we always schedule a transition and under the condition that no infinite internal computation is possible, the property above guarantees progress. However, in order to be sure that if $M_1 \sqsubseteq_D M_2$ and M_2 satisfies the property above then M_1 guarantee progress, we need to make sure that from every state of M_2 it is possible to avoid deadlock and there is no possibility of infinite internal computation. Such a property must be verified separately since it is not guaranteed by the trace distribution preorder. Fortunately, there are several cases (e.g., n processes running in parallel that communicate via shared memory) where it is easy to verify that it is always possible to avoid a deadlock.

To prove that the property above is preserved, suppose that $M_1 \sqsubseteq_D M_2$, and suppose that M_2 satisfies the the property above, while M_1 does not. Then there is a trace distribution of M_1 with infinite external computation where the probability of β' after β conditional to β is greater than p. Since $M_1 \sqsubseteq_D M_2$, there is a trace distribution of M_2 with infinite external computation where the probability of β' after β conditional to β is greater than p. This contradicts the hypothesis that M_2 satisfies the property above. ■

Example 7.3.3 The following property is preserved by the trace distribution preorder.

> "*In every computation where infinite external activity occurs with probability* 1*, if a finite trace β occurs, then, no matter what state is reached, a trace β' occurs ofter β with probability at least p.*"

A more concrete instantiation of the property above is "under the hypothesis that a distributed system never deadlocks, if a process has requested a service (β), then, no matter what state is reached, either the service has received a positive acknowledgment already (β'), or a positive acknowledgment will be received eventually with probability at least p". This property is preserved by the trace distribution preorder since it is equivalent to the property of Example 7.3.2 with $p = 1$ (cf. Proposition 5.5.5 to have an idea of why this is true). ■

Essentially, the rule of thumb to determine what properties can be guaranteed to be preserved under the trace distribution preorder is the following: express the property of interest as a property ϕ of the trace distributions of a probabilistic automaton M plus a condition ψ on the structure of M. If $M_1 \sqsubseteq_D M_2$, then the trace distributions of M_1 satisfy the property ϕ. Thus, if we know that M_2 satisfies the property of interest, it is enough to verify separately that M_1 satisfies ψ in order to be guaranteed that also M_1 satisfies the property of interest.

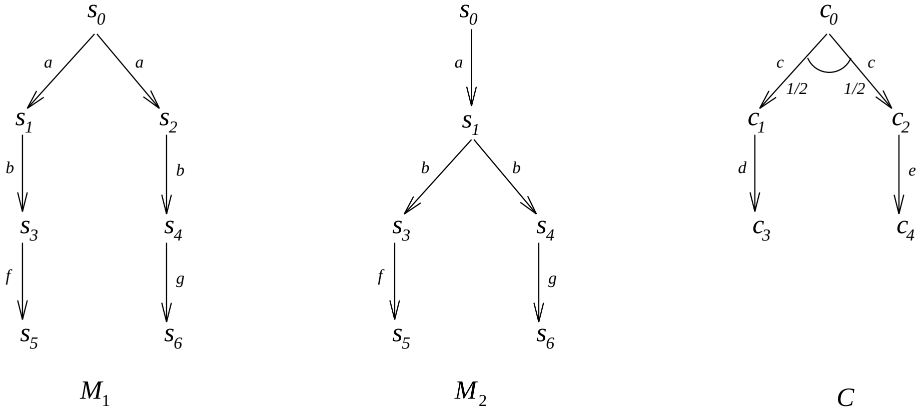


Figure 7-4: The trace distribution preorder is not a precongruence.

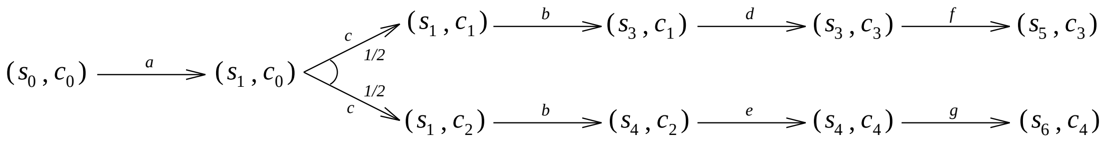


Figure 7-5: A probabilistic execution of $M_2 \| C$.

7.4 Trace Distribution Precongruence

Although the trace distribution preorder preserves some properties that are useful for the analysis of randomized distributed systems, the trace distribution preorder is not a precongruence, and thus it does not allow us to use modular analysis.

Example 7.4.1 (The trace distribution preorder is not substitutive) Consider the two probabilistic automata M_1 and M_2 of Figure 7-4. It is easy to check that M_1 and M_2 have the same trace distributions. Consider now the context C of Figure 7-4. Figure 7-5 shows a probabilistic execution of $M_2 \| C$ where there is a total correlation between the occurrence of actions d and f and actions e and g. Such a correlation cannot be obtained from $M_1 \| C$, since the choice between f and g must be resolved before knowing what action among d and e is chosen probabilistically. Thus, $M_1 \| C$ and $M_2 \| C$ do not have the same trace distributions. ■

This leads us to the definition of the *trace distribution precongruence*, denoted by $\sqsubseteq_{DC}$, as the coarsest precongruence that is contained in the trace distribution preorder. This definition of the trace distribution precongruence is not constructive, and thus it is difficult to understand what we have defined. Furthermore, we do not have any observational semantics that characterizes the trace distribution precongruence. In Section 7.5 we give an alternative characterization of the trace distribution precongruence that gives a better idea of the relation that we have defined. Here we give some examples of properties that are preserved by the trace distribution precongruence and that are not preserved by the trace distribution preorder.

Example 7.4.2 The following property is preserved by the trace distribution precongruence but not by the trace distribution preorder.

> "*After some finite trace β has occurred, no matter what state is reached, the probability that some other trace β' occurs from the state reached is not greater than p.*"

This property is not preserved by the trace distribution preorder since trace distributions cannot detect all the points where we may start to study the probability of β' to occur. However, this task is possible with the help of an external context. We use a context C that performs a fresh action o and then stops.

Suppose that $M_1 \sqsubseteq_{DC} M_2$ and suppose that M_2 satisfies the property above, while M_1 does not. Then there is a probabilistic execution H_1 of M_1 where some state q is reached after the occurrence of β, and the probability that β' occurs from q is greater than p. Consider a probabilistic execution H_1' of $M_1\|C$ such that $H_1'\lceil M_1 = H_1$ and such that action o is scheduled exactly from the minimal state q' such that $q'\lceil M_1 = q$. Then, o occurs always after β, and the conditional probability of β' after o given that o occurred is greater than p in the trace distribution of H_1'. Since $M_1 \sqsubseteq_{DC} M_2$, then there is a probabilistic execution H_2' of $M_2\|C$ whose trace distribution is the same as the trace distribution of H_2'. This means that there is at least one state q'' in H_2', reached immediately after the occurrence of o, where the probability that β' occurs from q'' in H_2' is greater than p. Consider $H_2'\lceil M_2$, and change its transition relation to obtain a probabilistic execution H_2 such that $H_2 \triangleright (q''\lceil M_2) = (H_2'\lceil M_2) \triangleright (q''\lceil M_2)$. Then the probability that β' occurs from $q''\lceil M_2$ in H_2 is greater than p. Moreover, β has occurred when $q\lceil M_2$ is reached. This contradicts the hypothesis that M_2 satisfies the property above. ■

Example 7.4.3 The following property is preserved by the trace distribution precongruence but not by the trace distribution preorder.

> "*In every computation where infinite external activity occurs with probability* 1*, if a finite trace β occurs, then, no matter what state is reached, if another trace β'' has not occurred yet after β, then a trace β' occurs with probability at least p.*"

A more concrete instantiation of the property above is "under the hypothesis that a distributed system never deadlocks, if a process has requested a service (β) and has not received yet a refusal (β'') then, no matter what state is reached, a positive acknowledgment (β') will be received eventually with probability at least p". Observe that the main difference from the property of Example 7.3.3 is in the use of β''. The presence of β'' does not guarantee that β' occurs with probability 1.

Even in this case in the proof we use a context C with a fresh action o. Suppose that $M_1 \sqsubseteq_{DC} M_2$ and suppose that M_2 satisfies the property above, while M_1 does not. Then there is a probabilistic execution H_1 of M_1 where infinite external activity occurs such that there is a state q of H_1 that is reached after the occurrence of β and before the occurrence of β'', and such that the probability that β' occurs from q is smaller than p. Consider a probabilistic execution H_1' of $M_1\|C$ such that $H_1'\lceil M_1 = H_1$ and such that action o is scheduled exactly from the minimal state q' such that $q'\lceil M_1 = q$. Then, o occurs always after β and before β'' occurs after β, and the conditional probability of β' after o given that o occurred is greater than p in the trace distribution of H_1'. Since $M_1 \sqsubseteq_{DC} M_2$, then there is a probabilistic execution H_2' of $M_2\|C$ whose trace distribution is the same as the trace distribution of H_2'. This means that there is at

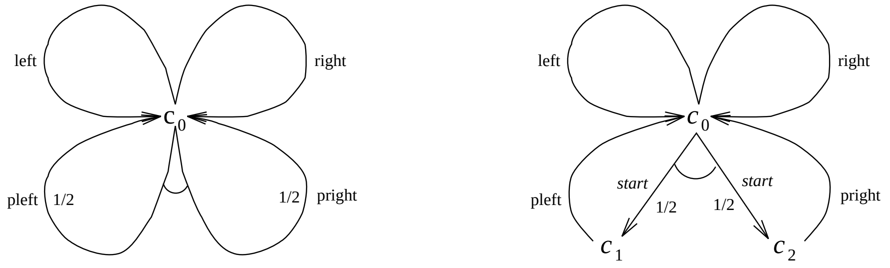


Figure 7-6: The principal context (left) and the simple principal context (right).

least one state q'' in H'_2, reached immediately after the occurrence of o, where the probability that β' occurs from q'' in H'_2 is smaller than p. Consider $H'_2\lceil M_2$, and change its transition relation to obtain a probabilistic execution H_2 such that $H_2 \triangleright (q''\lceil M_2) = (H'_2\lceil M_2) \triangleright (q''\lceil M_2)$. Then the probability that β' occurs from $q''\lceil M_2$ in H_2 is smaller than p. Moreover, β has occurred when $q\lceil M_2$ is reached and similarly β'' has not occurred after the occurrence of β. This contradicts the hypothesis that M_2 satisfies the property above. ■

7.5 Alternative Characterizations of the Trace Distribution Precongruence

In this section we give an alternative characterization of the trace distribution precongruence that is easier to manipulate. We define a *principal context*, denoted by C_P, and we show that there exists a context C that can distinguish two probabilistic automata M_1 and M_2 iff the principal context distinguishes M_1 and M_2.

7.5.1 The Principal Context

The principal context is a probabilistic automaton with a unique state and three self-loop transitions labeled with actions that do not appear in any other probabilistic automaton. Two self-loop transitions are deterministic (Dirac) and are labeled with action *left* and *right*, respectively; the third self-loop transition is probabilistic, where one edge leads to the occurrence of action *pleft* with probability 1/2 and the other edge leads to the occurrence of action *pright* with probability 1/2. Figure 7-6 shows the principal context.

The principal context is not a simple probabilistic automaton; however, since it does not have any action in common with any other probabilistic automaton, the parallel composition operator can be extended trivially: no synchronization is allowed. Alternatively, if we do not want a non-simple context, we can replace the principal context with the *simple principal context*, represented in Figure 7-6, as well. In this case we need to assume that also action *start* does not appear in any other probabilistic automaton. The main theorem is the following.

Theorem 7.5.1 $M_1 \sqsubseteq_{DC} M_2$ *iff* $M_1 \| C_P \sqsubseteq_D M_2 \| C_P$. ■

As a corollary we obtain an alternative characterization of the trace distribution precongruence and a compositional observational semantics for probabilistic automata. A *principal trace distri-*

bution of a probabilistic automaton M is a trace distribution of $M \| C_P$. Denote by $\mathit{ptdistrs}(M)$ the set $\mathit{tdistrs}(M \| C_P)$.

Corollary 7.5.2 *$M_1 \sqsubseteq_{DC} M_2$ iff $\mathit{ptdistrs}(M_1) \subseteq \mathit{ptdistrs}(M_2)$.* ■

The fact that the principal context is not a simple probabilistic automaton may appear to be confusing. Here we shed some light on the problem. First of all, in Chapter 4 we have defined parallel composition only for simple probabilistic automata; in this section, in order to account for the principal context, we have extended parallel composition to pairs of probabilistic automata, not necessarily simple, that do not have any action in common. This raises an immediate question: is the trace distribution precongruence defined based solely on contexts that are simple probabilistic automata or is it defined based on any compatible context according to the new extended parallel composition? The answer to this question, as it will become clear from the proof of Theorem 7.5.1, is that it does not matter because the two definitions are equivalent. That is, if there is a non-simple context that distinguishes two simple probabilistic automata M_1 and M_2, then the simple principal context distinguishes M_1 and M_2 as well.

Our choice of the principal context is just stylistic since it contains less structure than the simple principal context. The reader should keep in mind that there are infinitely many contexts with the same properties as the principal and the simple principal contexts; any one of those contexts can be chosen to give an alternative characterization to the trace distribution precongruence.

7.5.2 High Level Proof

The rest of this section is dedicated to the proof of Theorem 7.5.1. The proof is structured in several steps where at each step a generic distinguishing context C is transformed into a simpler distinguishing context C'. The proof of each transformation step is structured as follows. Given a distinguishing context C for $M_1 \sqsubseteq_D M_2$, build a simpler context C'. Suppose by contradiction that C' is not a distinguishing context and consider a trace distribution $\mathcal{D}$ of $M_1 \| C$ that is not a trace distribution of $M_2 \| C$. Let H_1 be a probabilistic execution of $M_1 \| C$ such that $\mathit{tdistr}(H_1) = \mathcal{D}$. Transform H_1 into a probabilistic execution H_1' of $M_1 \| C'$, and show that if there is a probabilistic execution H_2' of $M_2 \| C'$ such that $\mathit{tdistr}(H_2') = \mathit{tdistr}(H_1')$, then H_2' can be transformed into a probabilistic execution H_2 of $M_2 \| C$ such that $\mathit{tdistr}(H_2) = \mathcal{D}$. This leads to a contradiction.

The high level proof of Theorem 7.5.1 is then the following.

$\Longrightarrow$: Assuming that the principal context distinguishes M_1 and M_2, we show that the simple principal context distinguishes M_1 and M_2.

$\Longleftarrow$: We consider a generic context C that distinguishes M_1 and M_2, and we transform it into the principal context, showing that the principal context distinguishes M_1 and M_2. The transformation steps are the following.

1. Ensure that C does not have any action in common with M_1 and M_2 (Lemma 7.5.3);
2. Ensure that C does not have any cycles in its transition relation (Lemma 7.5.4);
3. Ensure that the branching structure of C is at most countable (Lemma 7.5.5);

4. Ensure that the branching structure of C is at most binary (Lemma 7.5.6);
5. Ensure that the probabilistic transitions of C lead to binary and uniform distributions (Lemma 7.5.7);
6. Ensure that each action of C is external and appears exactly in one edge of the transition relation of C (Lemma 7.5.8);
7. Ensure that each state of C enables two deterministic transitions and one probabilistic transition with a uniform binary distribution (Lemma 7.5.9);
8. Rename all the actions of the context of 7 according to the action names of the principal context and then collapse all the states of the new context into a unique state, leading to the principal context (Lemma 7.5.10).

7.5.3 Detailed Proof

Lemma 7.5.3 *Let C be a distinguishing context for two probabilistic automata M_1 and M_2. Then there exists a distinguishing context C' for M_1 and M_2 with no actions in common with M_1 and M_2. C' is called a* separated context.

Proof. The context C' is built from C be replacing each action a in common with M_1 and M_2, called a *shared* action, with two new actions a_1, a_2, and by replacing each transition $(c, a, \mathcal{P})$ of C with two transitions (c, a_1, c') and $(c', a_2, \mathcal{P})$, where c' denotes a new state that is used only for the transition $(c, a, \mathcal{P})$. We denote c' also by $c_{(c,a,\mathcal{P})}$ when convenient. We also denote the set of actions of the kind a_1 and a_2 by V_1 and V_2, respectively.

Let $\mathcal{D}$ be a trace distribution of $M_1\|C$ that is not a trace distribution of $M_2\|C$. Consider a probabilistic execution H_1 of $M_1\|C$ such that $tdistr(H_1) = \mathcal{D}$, and consider the scheduler that leads to H_1. Apply to $M_1\|C'$ the same scheduler with the following modification: whenever a transition $((s_1, c), a, \mathcal{P}_1 \otimes \mathcal{P})$ is scheduled in $M_1\|C$, schedule $((s_1, c), a_1, \mathcal{D}((s_1, c')))$, where c' is $c_{(c,a,\mathcal{P})}$, followed by $((s_1, c'), a, \mathcal{P}_1 \otimes \mathcal{D}(c'))$, and, for each $s_1' \in \Omega_1$, followed by $((s_1', c'), a_2, \mathcal{D}(s_1') \otimes \mathcal{P})$. Denote the resulting probabilistic execution by H_1' and the resulting trace distribution by $\mathcal{D}'$. Then,

$$\mathcal{D}' \restriction acts(M_1\|C) = \mathcal{D}. \tag{7.14}$$

To prove (7.14) we define a new construction, called *collapse* and abbreviated with *clp*, to be applied to probabilistic executions of $M_i\|C'$, $i = 1, 2$, where each occurrence of a shared action a is followed immediately by an occurrence of its corresponding action a_2.

Let H' be a probabilistic execution of $M_i\|C'$ where each occurrence of a shared action a is followed immediately by an occurrence of its corresponding action a_2. For convenience denote $clp(H')$ by H. A state q of H' is *closed* if each occurrence of a shared action a is followed eventually by an occurrence of the corresponding action a_2. For each closed state q of H', let $clp(q)$ be obtained from q as follows: each sequence

$$(s_0, c_0)a_1(s_0, c_{tr})\tau_2(s_2, c_{tr}) \cdots \tau_k(s_k, c_{tr})a(s, c_{tr})a_2(s, c)$$

is replaced with

$$(s_0, c_0)\tau_2(s_2, c_0) \cdots \tau_k(s_k, c_0)a(s, c),$$

and each sequence

$$(s_0, c_0)a_1(s_1, c_{tr})\tau_2(s_2, c_{tr})\cdots\tau_k(s_k, c_{tr})$$

occurring at the end of q is replaced with

$$(s_0, c_0)\tau_2(s_2, c_0)\cdots\tau_k(s_k, c_0).$$

Define

$$states(H) \triangleq \{clp(q) \mid q \in states(H'), closed(q)\}. \tag{7.15}$$

Let $(q, \mathcal{P})$ be a restricted transition of H' where q is a closed state, and suppose that no action of $V_1 \cup V_2$ occurs. Consider a pair (a, q') of Ω. If a is not a shared action, then let

$$\mathcal{P}_{(a,q')} \triangleq \mathcal{D}((a, clp(q'))); \tag{7.16}$$

if a is a shared action, then let

$$\Omega_{(a,q')} \triangleq \{(a, clp(q'')) \mid (a_2, q'') \in \Omega_{q'}^{H'}\}, \tag{7.17}$$

and for each $(a, q''') \in \Omega_{(a,q')}$, let

$$P_{(a,q')}[(a, q''')] \triangleq P_{q'}[a_2 \times clp^{-1}(q''')], \tag{7.18}$$

where for each state q of H, $clp^{-1}(q)$ is the set of closed states q' of H' such that $clp(q') = q$. The transition $clp((q, \mathcal{P}))$ is defined to be

$$clp((q, \mathcal{P})) \triangleq \left(clp(q), \sum_{(a,q')\in\Omega} P[(a, q')]\mathcal{P}_{(a,q')}\right). \tag{7.19}$$

For the transition relation of H, consider a state q of H Let $min(clp^{-1}(q))$ be the set of minimal states of $clp^{-1}(q)$ under prefix ordering. For each state $\bar{q} \in clp^{-1}(q)$, let

$$\bar{p}_{\bar{q}}^{clp^{-1}(q)} \triangleq \frac{P_{H'}[C_{\bar{q}}]}{\sum_{q'\in min(clp^{-1}(q))} P_{H'}[C_{q'}]}. \tag{7.20}$$

The transition enabled in H from q is

$$\sum_{q'\in clp^{-1}(q)} \bar{p}_{q'}^{clp^{-1}(q)} P_{q'}^{H'}[acts(M_i\|C)]clp(tr_{q'}^{H'} \upharpoonright acts(M_i\|C)). \tag{7.21}$$

Note the similarity with the definition of the projection of a probabilistic execution fragment (cf. Section 4.3.2).

The probabilistic execution H satisfies the following properties.

a. H is a probabilistic execution of $M_i\|C$.

   The fact that each state of H is reachable can be shown by a simple inductive argument; the fact that each state of H is a finite execution fragment of $M_i\|C$ follows from a simple analysis of the definition of clp.

   From (7.21) it is enough to check that for each closed state q' of H', the transition $clp(tr_{q'}^{H'} \upharpoonright acts(M_i\|C))$ is generated by a combination of transitions of $M_i\|C$. Since $tr_{q'}^{H'}$ is a transition of H', $(tr_{q'}^{H'} \upharpoonright acts(M_i\|C))$ can be expressed as $\sum_j p_j(q' \frown tr_j)$, where each tr_j is a transition of $M_i\|C'$. We distinguish three cases.

1. tr_j is a non-shared transition of M_i.

   Then $tr_j = ((s,c), a, \mathcal{P} \otimes \mathcal{D}(c))$ for some action a and probability space $\mathcal{P}$, where $(s,c) = \mathit{lstate}(q')$. Let $\mathit{lstate}(\mathit{clp}(q')) = (s', c')$. Then, $s' = s$, as it follows directly from the definition of clp. Define tr'_j to be the transition $((s,c'), a, \mathcal{P} \otimes \mathcal{D}(c'))$. Then tr'_j is a transition of $M_i \| C$ and $\mathit{clp}(q' \frown tr_j) = \mathit{clp}(q') \frown tr'_j$

2. tr_j is a non-shared transition of C'.

   Then $tr_j = ((s,c), a, \mathcal{D}(s) \otimes \mathcal{P})$ for some action a and probability space $\mathcal{P}$, where $(s,c) = \mathit{lstate}(q')$. Let $\mathit{lstate}(\mathit{clp}(q')) = (s', c')$. Then, $s' = s$ and $c' = c$, as it follows directly from the definition of clp after observing that q' must be a closed state in order to enable tr_j. Define tr'_j to be tr_j. Then tr'_j is a transition of $M_i \| C$ and $\mathit{clp}(q' \frown tr_j) = \mathit{clp}(q') \frown tr'_j$

3. tr_j is a shared transition.

   Then $tr_j = ((s, c_{tr}), a, \mathcal{P} \otimes \mathcal{D}(c_{tr}))$ for some action a and probability space $\mathcal{P}$, where $(s, c_{tr}) = \mathit{lstate}(q')$. In particular, c_{tr} is one of the states that are added to those of C, and tr is a simple transition of C with action a. Moreover, from each state $(s', c_{tr}) \in \Omega_{\mathcal{P} \otimes \mathcal{D}(c_{tr})}$, there is a transition $((s', c_{tr}), a_2, \mathcal{D}(s') \otimes \mathcal{P}_{tr})$ enabled. Let $\mathit{lstate}(\mathit{clp}(q')) = (s', c')$. Then, $s' = s$. Define tr'_j to be $((s, c'), a, \mathcal{P} \otimes \mathcal{P}_{tr})$. Then, from the definition of C', tr'_j is a transition of $M_i \| C$.

Observe that clp distributes over combination of transitions. Moreover, from Equation (7.19), observe that for each j $\mathit{clp}(q' \frown tr_j) = \mathit{clp}(q') \frown tr'_j$. Thus, $\mathit{clp}(tr^{H'}_{q'} \restriction \mathit{acts}(M_i \| C)) = \mathit{clp}(q') \frown (\sum_j p_j tr'_j)$, which is generated by a combination of transitions of $M_i \| C$.

b. For each state q of H,

$$P_H[C_q] = \sum_{q' \in \mathit{min}(\mathit{clp}^{-1}(q))} P_{H'}[C_{q'}]. \tag{7.22}$$

This is shown by induction on the length of q. If q consists of a start state only, then the result is trivial. Otherwise, from the definition of the probability of a cone, Equation (7.21), and a simple algebraic simplification,

$$P_H[C_{qas}] = P_H[C_q] \left(\sum_{q' \in \mathit{clp}^{-1}(q)} \bar{p}_{q'}^{\mathit{clp}^{-1}(q)} F_{q'}(qas) \right), \tag{7.23}$$

where $F_{q'}(qas)$ expresses the probability of the completions of q' to a state whose collapse gives qas without using actions from $V_1 \cup V_2$ in the first transition. Formally, if a is not a shared action, then $F_{q'}(qas)$ is $P^{H'}_{q'}[a \times \mathit{clp}^{-1}(qas)]$; otherwise, $F_{q'}(qas)$ is $P^{H'}_{q'}[(a, q'a(s', c_{tr}))] P^{H'}_{q'a(s', c_{tr})}[(a_2, q'a(s', c_{tr})a_2(s', c))]$, where $c_{tr} = \mathit{lstate}(q') \lceil C'$, and $s = (s', c)$. In the first case, $\Omega^{H'}_{q'} \cap (\{a\} \times \mathit{clp}^{-1}(qas))$ contains only one element, say $(a, q'as'')$, and $P_{H'}[C_{q'}] F_{q'}(qas)$ gives $P_{H'}[C_{q'as''}]$; in the second case $P_{H'}[C_{q'}] F_{q'}(qas)$ gives $P_{H'}[C_{(q'a(s', c_{tr})a_2 s)}]$.

Observe that the states of $min(clp^{-1}(qas))$ are the states of the form described above (simple cases analysis). Thus, by applying induction to (7.23), using (7.20), simplifying algebraically, and using the observations above,

$$P_H[C_{qas}] = \sum_{q' \in min(clp^{-1}(qas))} P_{H'}[C_{q'}]. \tag{7.24}$$

c. $tdistr(H) = tdistr(H') \upharpoonright acts(M_i \| C)$.

Let β be a finite trace of H or H'. Then $\{\alpha \in \Omega_{H'} \mid \beta \leq trace(\alpha) \upharpoonright acts(M_i \| C)\}$ can be expressed as a union of disjoint cones $\cup_{q \in \Theta} C_q$ where, if the last action of β is a and a is not a shared action,

$$\Theta = \{q \in states(H') \mid trace(q) \upharpoonright acts(M_i \| C) = \beta, lact(q) = a\}, \tag{7.25}$$

and if the last action of β is a and a is a shared action,

$$\Theta = \{q \in states(H') \mid trace(q) \upharpoonright acts(M_i \| C) = \beta, lact(q) = a_2\}. \tag{7.26}$$

Observe that Θ is a set of closed states. The set $clp(\Theta)$ is the set

$$clp(\Theta) = \{q \in states(H) \mid trace(q) = \beta, lact(q) = a\}, \tag{7.27}$$

which is a characterization of $\{\alpha \in \Omega_H \mid \beta \leq trace(\alpha)\}$ as a union of disjoint cones. Observe that $min(clp^{-1}(clp(\Theta))) = \Theta$. Moreover, for each $q_1 \neq q_2$ of $clp(\Theta)$, $clp^{-1}(q_1) \cap clp^{-1}(q_2) = \emptyset$. Thus, from (7.22), $P_{H'}[\cup_{q \in \Theta} C_q] = P_H[\cup_{q \in clp(\Theta)} C_q]$. This is enough to conclude.

To complete the proof of (7.14) it is enough to observe that $H_1 = clp(H_1')$. Property (7.14) is then expressed by property (c).

Suppose by contradiction that it is possible to obtain $\mathcal{D}'$ from $M_2 \| C'$. Consider the scheduler that leads to $\mathcal{D}'$ in $M_2 \| C'$, and let H_2' be the corresponding probabilistic execution. First, we build a new probabilistic execution H_2'' of $M_2 \| C'$ whose trace distribution is $\mathcal{D}'$, and such that each shared action a is followed immediately by its corresponding action a_2. Then we let H_2 be $clp(H_2'')$. This leads to a contradiction since $tdistr(H_2) = \mathcal{D}$. The rest of the proof is dedicated to the construction of H_2''.

For each state q of H_2', let $exch(q)$ be the set of sequences q' that can be obtained from q as follows: each sequence

$$(s_0, c_{tr}) a (s_1, c_{tr}) \tau_2 (s_2, c_{tr}) \cdots \tau_h (s_h, c_{tr}) a_2 (s_h, c)$$

is replaced with

$$(s_0, c_{tr}) a (s_1, c_{tr}) a_2 (s_1, c) \tau_2 (s_2, c) \cdots \tau_h (s_h, c),$$

each sequence

$$(s_0, c_{tr}) a (s_1, c_{tr}) \tau_2 (s_2, c_{tr}) \cdots \tau_h (s_h, c_{tr})$$

occurring at the end of q is replaced with

$$(s_0, c_{tr})a(s_1, c_{tr})a_2(s_1, c)\tau_2(s_2, c)\cdots\tau_h(s_h, c),$$

where c is any of the states that a_2 may lead to from c_{tr}, and each sequence

$$(s_0, c_{tr})a(s_1, c_{tr})$$

occurring at the end of q, where a is a shared action, either it is replaced with

$$(s_0, c_{tr})a(s_1, c_{tr})a_2(s_1, c),$$

where c is any of the states that a_2 may lead to from c_{tr}, or it is not replaced. Then, define

$$states(H_2'') \triangleq \bigcup_{q\in states(H_2')} exch(q). \tag{7.28}$$

Let $(q, \mathcal{P})$ be a restricted transition of H_2', and suppose that no action of V_2 occurs. Let q' be a state of $exch(q)$ that does not end with a shared action. Then, for each $(a, q_1) \in \Omega$ there is exactly one $q_1' \in exch(q_1)$ such that $q' \leq q_1'$ and $|q_1'| = |q'| + 1$ (simple analysis of the definition of $exch$). Denote such q_1' by $exch_{q'}(q_1)$. Let $\Omega' = \{(a, exch_{q'}(q_1) \mid (a, q_1) \in \Omega\}$, and let, for each $(a, q_1') \in \Omega'$, $P'[(a, q_1')] = P[(a \times exch^{-1}(q_1'))]$, where $exch^{-1}(q)$ is the set of states q' of H_2' such that $q \in exch(q')$. Then define the transition $exch_{q'}((q, \mathcal{P}))$ to be

$$exch_{q'}((q, \mathcal{P})) \triangleq (q', \mathcal{P}'). \tag{7.29}$$

For each state q of H_2'', let $min(exch^{-1}(q))$ be the set of minimal states of $exch^{-1}(q)$ under prefix ordering. For each state q' of $exch^{-1}(q)$, where q is closed, let

- $p_{q'}^q \triangleq P_{H_2'}[C_{q'}]$ if q' is *closed*, i.e., if each occurrence of a shared action a is followed eventually by an occurrence of its corresponding action a_2;
- $p_{q'}^q \triangleq P_{H_2'}[C_{q'}]P_{tr}[c]$ if q' is open, where $lstate(q')\lceil C' = c_{tr}$ and $lstate(q)\lceil C = c$.

For each $q' \in exch^{-1}(q)$, let

$$\bar{p}_{q'}^{exch^{-1}(q)} \triangleq \frac{p_{q'}^q}{\sum_{q''\in min(exch^{-1}(q))} p_{q''}^q}. \tag{7.30}$$

If the last action of q is a shared action a, and $lstate(q) = (s, c_{tr})$, then the transition enabled from q in H_2'' is

$$q \frown ((s, c_{tr}), a_2, \mathcal{D}(s) \otimes \mathcal{P}_{tr}). \tag{7.31}$$

If the last action of q is not a shared action, then the transition enabled from q in H_2'' is

$$\sum_{q'\in exch^{-1}(q)} \bar{p}_{q'}^{exch^{-1}(q)} P_{q'}^{H_2'}[acts(H_2')\backslash V_2] exch_q(tr_{q'}^{H_2'} \upharpoonright (acts(H_2')\backslash V_2)). \tag{7.32}$$

The probabilistic execution H_2' satisfies the following properties.

a. H_2'' is a probabilistic execution of $M_2\|C'$.

The fact that each state of H_2'' is reachable can be shown by a simple inductive argument; the fact that each state of H_2'' is a finite execution fragment of $M_2\|C'$ follows from a simple analysis of the definition of *exch*.

We need to check that for each state q of H_2'' the transition enabled from q in H_2'' is generated by a combination of transitions of $M_2\|C'$. If the last action of q is a shared action, then the result follows immediately from Expression (7.31) and the definition of C'. If the last action of q is not a shared action, then consider a state $q' \in exch^{-1}(q)$. The transition $tr_{q'}^{H_2'} \restriction (acts(H_2')\backslash V_2)$ can be expressed as $\sum_i p_i(q' \frown tr_i)$, where each tr_i is a transition of $M_2\|C'$ enabled from $lstate(q')$. We distinguish three cases.

1. tr_i is a non-shared transition of M_2.
   Then $tr_i = ((s,c), a, \mathcal{P} \otimes \mathcal{D}(c))$ for some action a and probability space $\mathcal{P}$, where $(s,c) = lstate(q')$. Let $lstate(q) = (s',c')$. Then, $s' = s$. Define tr_i' to be the transition $((s,c'), a, \mathcal{P}\otimes\mathcal{D}(c'))$. Then tr_i' is a transition of $M_2\|C'$ and $exch_q(q' \frown tr_i) = q \frown tr_i'$.
2. tr_i is a non-shared transition of C'.
   Then $tr_i = ((s,c), a, \mathcal{D}(s) \otimes \mathcal{P})$ for some action a and probability space $\mathcal{P}$, where $(s,c) = lstate(q')$. Let $lstate(q) = (s',c')$. Then, $s' = s$ and $c = c'$. Define tr_i' to be tr_i. Then tr_i' is a transition of $M_2\|C'$ and $exch_q(q' \frown tr_i) = q \frown tr_i'$.
3. tr_i is a shared transition.
   Then $tr_i = ((s,c), a, \mathcal{P} \otimes \mathcal{D}(c))$ for some action a and probability space $\mathcal{P}$, where $(s,c) = lstate(q')$. Let $lstate(q) = (s',c')$. Then, $s' = s$ and $c = c'$. Define tr_i' to be tr_i. Then tr_i' is a transition of $M_2\|C'$ and $exch_q(q' \frown tr_i) = q \frown tr_i'$.

Observe that *exch* distributes over combination of transitions. Thus, $exch_q((tr_{q'}) \restriction (acts(H_2')\backslash V_2))$ can be expressed as $\sum_i p_i(q \frown tr_i')$, which is generated by a combination of transitions of $M_2\|C'$. From (7.32), the transition enabled from q in H_2'' is generated by a combination of transitions of $M_2\|C'$.

b. For each state q of H_2'',

$$P_{H_2''}[C_q] = \begin{cases} \sum_{q' \in min(exch^{-1}(q))} P_{H_2'}[C_{q'}] & \text{if } q \text{ ends with a shared action,} \\ \sum_{q' \in min(exch^{-1}(q))} p_{q'}^{q} & \text{otherwise.} \end{cases} \tag{7.33}$$

The proof is by induction on the length of q. If q consists of a start state only, then the result is trivial. Otherwise, consider $P_{H_2''}[C_{qas}]$. We distinguish two cases.

1. q is open.
   In this case, since in H_2' each shared action is followed immediately by the corresponding action of V_2, a is an action of V_2. Moreover, from the definition of *exch*,

   $$exch^{-1}(q) = min(exch^{-1}(qas)) = min(exch^{-1}(q)), \tag{7.34}$$

   and all the elements of $exch^{-1}(q)$ are open states. From induction,

   $$P_{H_2''}[C_q] = \sum_{q' \in min(exch^{-1}(q))} P_{H_2'}[C_{q'}]. \tag{7.35}$$

Let $c = s\lceil M_2$, and let $c_{tr} = \mathit{lstate}(q)\lceil C'$. Then, for each $q' \in \mathit{min}(\mathit{exch}^{-1}(q))$, $c_{tr} = \mathit{lstate}(q')\lceil C'$, and

$$p_{q'}^{qas} = P_{H_2'}[C_{q'}]P_{tr}[c]. \tag{7.36}$$

Moreover, $P_q^{H_2''}[(a, qas)] = P_{tr}[c]$. Thus, from the definition of the probability of a cone and (7.35),

$$P_{H_2''}[C_{qas}] = \sum_{q' \in \mathit{min}(\mathit{exch}^{-1}(q))} P_{H_2'}[C_{q'}]P_{tr}[c]. \tag{7.37}$$

By using the fact that $\mathit{min}(\mathit{exch}^{-1}(q)) = \mathit{min}(\mathit{exch}^{-1}(qas))$, and using (7.36), we obtain

$$P_{H_2''}[C_{qas}] = \sum_{q' \in \mathit{min}(\mathit{exch}^{-1}(qas))} p_{q'}^{qas}. \tag{7.38}$$

2. q is closed.

   In this case, from the definition of the probability of a cone and (7.32),

   $$P_{H_2''}[C_{qas}] = P_{H_2''}[C_q]\left(\sum_{q' \in \mathit{exch}^{-1}(q)} \bar{p}_{q'}^{\mathit{exch}^{-1}(q)} P_{q'}^{H_2'}[a \times \mathit{exch}^{-1}(qas)]\right). \tag{7.39}$$

   Let $Ptr_q[q']$ denote $P_{tr}[c]$, where $c = \mathit{lstate}(q)\lceil C'$, and $c_{tr} = \mathit{lstate}(q')\lceil C'$. Then, from induction and (7.30),

   $$\begin{aligned} P_{H_2''}[C_{qas}] \;=\; & \sum_{q' \in \mathit{exch}^{-1}(q) \mid \mathit{closed}(q')} P_{H_2'}[C_{q'}]P_{q'}^{H_2'}[a \times \mathit{exch}^{-1}(qas)] + \\ & \sum_{q' \in \mathit{exch}^{-1}(q) \mid \mathit{open}(q')} P_{H_2'}[C_{q'}]Ptr_q[q']P_{q'}^{H_2'}[a \times \mathit{exch}^{-1}(qas)]. \end{aligned} \tag{7.40}$$

   We distinguish two subcases.

   (a) a is a shared action.

      In this case each state q' of $\mathit{exch}^{-1}(q)$ such that $P_{q'}^{H_2'}[a \times \mathit{exch}^{-1}(qas)] > 0$ is closed. Thus, only the first summand of (7.40) is used. Moreover, each state of $\mathit{min}(\mathit{exch}^{-1}(qas))$ is captured by Expression (7.40). Thus, $P_{H_2'}[C_{qas}] = \sum_{q' \in \mathit{min}(\mathit{exch}^{-1}(qas))} P_{H_2'}[C_{q'}]$. Observe that qas is open.

   (b) a is not a shared action.

      In this case, for each $q' \in \mathit{exch}^{-1}(q)$, if q' is closed, then all the states reached in $\Omega_{q'} \cap (\{a\} \times \mathit{exch}^{-1}(qas))$ are closed, and if q' is open, then all the states reached in $\Omega_{q'} \cap (\{a\} \times \mathit{exch}^{-1}(qas))$ are open. Moreover, each state of $\mathit{min}(\mathit{exch}^{-1}(qas))$ is captured by Expression (7.40). Thus, from the definition of $p_{q'}^{qas}$, $P_{H_2'}[C_{qas}] = \sum_{q' \in \mathit{min}(\mathit{exch}^{-1}(qas))} p_{q'}^{qas}$. Observe that qas is closed.

c. $\mathit{tdistr}(H_2') = \mathit{tdistr}(H_2'')$.

Let β be a finite trace of H_2' or H_2''. Then $\{\alpha \in \Omega_{H_2'} \mid \beta \leq \mathit{trace}(\alpha)\}$ can be expressed as a union of disjoint cones $\cup_{q \in \Theta} C_q$ where

$$\Theta = \{q \in \mathit{states}(H') \mid \mathit{trace}(q) = \beta, \mathit{lact}(q) = \mathit{lact}(\beta)\}. \tag{7.41}$$

We distinguish two cases.

1. β does not end with an action of V_2.

   The set $\Theta' = \{q \in \mathit{exch}(\Theta) \mid \mathit{lact}(q) = \mathit{lact}(\beta)\}$ is a characterization of $\{\alpha \in \Omega_{H_2''} \mid \beta \leq \mathit{trace}(\alpha)\}$ as a union of disjoint cones. Observe that $\mathit{min}(\mathit{exch}^{-1}(\Theta')) = \Theta$ and that for each pair of states $q_1 \neq q_2$ of Θ', $\mathit{min}(\mathit{exch}^{-1}(q_1)) \cap \mathit{min}(\mathit{exch}^{-1}(q_2)) = \emptyset$. Thus, if β ends with a shared action, then (7.33) is sufficient to conclude that $P_{H_2'}[\{\alpha \in \Omega_{H_2'} \mid \beta \leq \mathit{trace}(\alpha)\}] = P_{H_2''}[\{\alpha \in \Omega_{H_2''} \mid \beta \leq \mathit{trace}(\alpha)\}]$; if β does not end with a shared action, then, since all the states of Θ are closed, Equation (7.33) together with the definition of $p_{q'}^{q}$ are sufficient to conclude.

2. β ends with an action of V_2.

   In this case $\beta = \beta' a_2$ for some action $a_2 \in V_2$. Observe that, both in H_2' and H_2'', after the occurrence of a shared action a the corresponding action a_2 occurs with probability 1: for H_2' recall that $\mathit{tdistr}(H_2') \upharpoonright \mathit{acts}(M_2 \| C) = \mathcal{D}$; for H_2'' see (7.31). Thus, the probability of β is the same as the probability of β', and the problem is reduced to Case 1. ■

Lemma 7.5.4 *Let C be a distinguishing separated context for two probabilistic automata M_1 and M_2. Then there exists a distinguishing cycle-free separated context C' for M_1 and M_2.*

Proof. C' can be built by unfolding C. Every scheduler for $M_i \| C$ can be transformed into a scheduler for $M_i \| C'$ and vice versa, leading to the same trace distributions. ■

Lemma 7.5.5 *Let C be a distinguishing cycle-free, separated context for two probabilistic automata M_1 and M_2. Then there exists a distinguishing cycle-free separated context C' for M_1 and M_2 with a transition relation that is at most countably branching.*

Proof. Let $\mathcal{D}$ be a trace distribution of $M_1 \| C$ that is not a trace distribution of $M_2 \| C$. Consider the corresponding probabilistic execution H. Observe that H has at most countably many states, and that at each state of H there are at most countably many transitions of C that are scheduled. Thus, in total, only countably many transitions of C are used to generate $\mathcal{D}$. Then C' is C without the unused transitions. ■

Lemma 7.5.6 *Let C be a distinguishing cycle-free, separated context for two probabilistic automata M_1 and M_2 such that the transition relation of C is at most countably branching. Then there exists a distinguishing cycle-free separated context C' for M_1 and M_2 that at each state either enables two deterministic transitions or a unique probabilistic transition with two possible outcomes. C' is called a* binary separated context*.*

Proof. For each state s of C, choose a new action start_s. Let s enable the transitions $\mathit{tr}_1, \mathit{tr}_2, \ldots$, where each tr_i is a transition $(s, a_i, \mathcal{P}_i)$. The transition relation of C' is obtained in two phases. First, a transition is chosen nondeterministically as shown in Figure 7-7, where each symbol $\bullet$ denotes a distinct state and each symbol τ denotes a distinct internal action; then, for each state $\bullet_i$, the transition tr_i is encoded as follows. Let Ω_i be $\{s_{i,1}, s_{i,2}, \ldots\}$, $p_{i,j} \triangleq P_i[s_{i,j}]$, and $\bar{p}_{i,j} \triangleq \sum_{k \geq j} p_{i,k}$. The transition relation from $\bullet_i$ is represented in Figure 7-8, where each

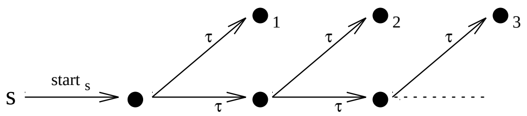


Figure 7-7: Nondeterministic choice of a transition.

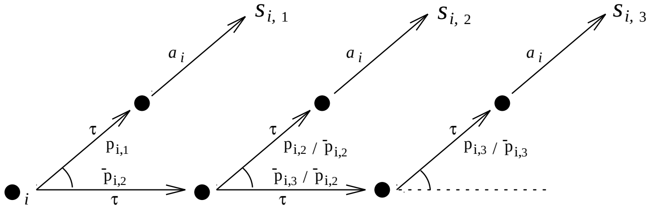

$s_{i,1}$
$s_{i,2}$
$s_{i,3}$
a_i
a_i
a_i
τ
τ
τ
$p_{i,1}$
$p_{i,2} / \bar{p}_{i,2}$
$p_{i,3} / \bar{p}_{i,3}$
$\bar{p}_{i,2}$
$\bar{p}_{i,3} / \bar{p}_{i,2}$
τ
τ
i

Figure 7-8: Transforming a transition into binary transitions.

symbol $\bullet$ denotes a distinct state and each symbol τ denotes a distinct internal action. Observe that by scheduling all the transitions of the diagram above, for each j we have

$$P[s_{i,j}] = P_i[s_{i,j}], \tag{7.42}$$

where $P[s_{i,j}]$ is the probability of reaching $s_{i,j}$ from $\bullet_i$. Denote the set of actions of the kind $start_s$ by V_{start}. Denote the auxiliary actions of C' that occur between a *start* action and a state $\bullet_j$ by V_1, and denote the auxiliary actions of C' that occur between a state $\bullet_j$ and the corresponding occurrence of action a_j by V_2.

Let $\mathcal{D}$ be a trace distribution of $M_1 \| C$ that is not a trace distribution of $M_2 \| C$. Consider a probabilistic execution H_1 of $M_1 \| C$ whose trace distribution is $\mathcal{D}$ in $M_1 \| C$, and consider the scheduler that leads to H_1 in $M_1 \| C$. Apply to $M_1 \| C'$ the same scheduler with the following modification: whenever some transition of C is scheduled, schedule the *start* action from C', then schedule the internal transitions to choose the transition of C to perform with the right probability, and then schedule the transitions of the chosen transition till the corresponding external action of C occurs. Denote the resulting probabilistic execution by H_1' and the resulting trace distribution by $\mathcal{D}'$. Then,

$$\mathcal{D}' \upharpoonright acts(M_1 \| C) = \mathcal{D}. \tag{7.43}$$

To prove (7.43), we define a new construction, called *shrink* and abbreviated with *shr*, to be applied to probabilistic executions of $M_i \| C'$ such that no action of M_i occurs between a state of the form $\bullet_j$ and the occurrence of the corresponding action a_j of C, and such that all the transitions between a state of the kind $\bullet_j$ and the corresponding occurrences of action a_j are scheduled.

Let H' be such a probabilistic execution of $M_i \| C'$. Denote $shr(H')$ by H. A state q of H' is *closed* if each occurrence of a state of the kind $\bullet_j$ is followed eventually by the occurrence of the corresponding action a_j. For each state q of H' let $shr(q)$ be obtained from q as follows: each sequence

$$(s_0, c_0) start_{c_0} (s_0, \bullet) b_1 (s_1, \bullet) \cdots b_h (s_h, \bullet_j) \tau_1 (s_h, \bullet) \cdots \tau_k (s_h, \bullet) a_j (s, c)$$

is replaced with

$$(s_0, c_0)b_{i_1}(s_{i_1}, c_0) \cdots b_{i_l}(s_{i,l}, c_0)a_j(s, c),$$

where $i_1, \ldots, i_l$ is the ordered sequence of the indexes of the b's that are actions of M_i, and each sequence either of the form

$$(s_0, c_0)start_{c_0}(s_0, \bullet)b_1(s_1, \bullet) \cdots b_h(s_h, \bullet_j)\tau_1(s_h, \bullet) \cdots \tau_k(s_h, \bullet)$$

or of the form

$$(s_0, c_0)start_{c_0}(s_0, \bullet)b_1(s_1, \bullet) \cdots b_h(s_h, \bullet)$$

occurring at the end of q is replaced with

$$(s_0, c_0)b_{i_1}(s_{i_1}, c_0) \cdots b_{i_l}(s_{i,l}, c_0),$$

where $i_1, \ldots, i_l$ is the ordered sequence of the indexes of the b's that are actions of M_i. Then,

$$states(H) \triangleq \{shr(q) \mid q \in states(H')\}. \tag{7.44}$$

Let $(q, \mathcal{P})$ be a restricted transition of H', and suppose that no action of $acts(C') \backslash acts(C)$ occurs. Let $\Omega' = \{(a, shr(q')) \mid (a, q') \in \Omega\}$, and for each $(a, q'') \in \Omega'$, let $P'[(a, q'')] = P[a \times shr^{-1}(q'')]$, where $shr^{-1}(q)$ is the set of states q' of H' such that $shr(q') = q$. Then the transition $shr((q, \mathcal{P}))$ is defined to be

$$shr((q, \mathcal{P})) \triangleq (shr(q), \mathcal{P}). \tag{7.45}$$

For the transition relation of H, consider a state q of H, and let $min(shr^{-1}(q))$ be the set of minimal states of $shr^{-1}(q)$ under prefix ordering. For each state $\bar{q} \in shr^{-1}(q)$, let

$$\bar{p}_{\bar{q}}^{shr^{-1}(q)} \triangleq \frac{P_{H'}[C_{\bar{q}}]}{\sum_{q' \in min(shr^{-1}(q))} P_{H'}[C_{q'}]}. \tag{7.46}$$

The transition enabled from q in H is

$$\sum_{q' \in shr^{-1}(q)} \bar{p}_{\bar{q}}^{shr^{-1}(q)} P_{q'}^{H'}[acts(M_i \| C)] shr(tr_{q'}^{H'} \restriction acts(M_i \| C)). \tag{7.47}$$

The probabilistic execution H satisfies the following properties.

a. H is a probabilistic execution of $M_i \| C$.

   The fact that each state of H is reachable can be shown by a simple inductive argument; the fact that each state of H is a finite execution fragment of $M_i \| C$ follows from a simple analysis of the definition of shr.

   We need to show that for each state q of H the transition of Expression (7.47) is generated by a combination of transitions of $M_i \| C$. The states of $shr^{-1}(q)$ that enable some action of $M_i \| C$ can be partitioned into two sets Θ_c and Θ_o of closed and open states, respectively.

   We analyze Θ_c first. Let $q' \in \Theta_c$. Since $tr_{q'}$ is a transition of H', $(tr_{q'} \restriction acts(M_i \| C))$ can be expressed as $\sum_j p_j(q' \frown tr_j)$, where each tr_j is a transition of $M_i \| C'$. We distinguish two cases.

1. tr_j is a transition of M_i.

   Then $tr_j = ((s,c), a, \mathcal{P} \otimes \mathcal{D}(c))$ for some action a and probability space $\mathcal{P}$, where $(s,c) = \mathit{lstate}(q')$. Let $\mathit{lstate}(\mathit{shr}(q')) = (s', c')$. Then, $s' = s$, as it follows directly from the definition of shr. Moreover, $(s, a, \mathcal{P})$ is a transition of M_i. Define tr'_j to be the transition $((s,c'), a, \mathcal{P} \otimes \mathcal{D}(c'))$. Then tr'_j is a transition of $M_i \| C$ and $\mathit{shr}_q(q' \frown tr_i) = q \frown tr'_j$.
2. tr_j is a transition of C'.

   This case is not possible since, from the construction of C', no action of C can be enabled from a closed state.

Observe that shr distributes over combination of transitions. Thus,

$$\mathit{shr}(tr_{q'}^{H'} \restriction \mathit{acts}(M_i\|C)) = \sum_j p_j(\mathit{shr}(q') \frown tr'_j), \tag{7.48}$$

which is generated by a combination of transitions of $M_i\|C$.

We now turn to Θ_o. The set Θ_o can be partitioned into sets $(\Theta_j)_{j\geq 0}$, where each set Θ_j consists of those states q' of Θ_o where a particular state $\bullet_j$ of C' occurs without its matching action a_j. Each element q' of Θ_j can be split into two parts $q_1 \frown q_2$, where $\mathit{lstate}(q_1)\lceil C' = \bullet_j$. Denote q_1 by $\mathit{head}(q')$. Partition Θ_j into other sets $(\Theta_{j,k})_{k\geq 0}$, where each $\Theta_{j,k}$ is an equivalence class of the relation that relates two states iff they have the same head. Denote the common head of the states of $\Theta_{i,j}$ by $\mathit{head}(\Theta_{i,j})$. For each pair of states q_1, q_2 of H' such that $q_1 \leq q_2$, denote by $p_{q_1q_2}$ the probability value such that $P_{H'}[C_{q_2}^{H'}] = P_{H'}[C_{q_1}^{H'}]p_{q_1q_2}$. Then, for each equivalence class $\Theta_{i,j}$, the expression

$$\sum_{q'\in\Theta_{j,k}} \bar{p}_{q'}^{\mathit{shr}^{-1}(q)} P_{q'}^{H'}[\mathit{acts}(M_i\|C)]\mathit{shr}(tr_{q'}^{H'} \restriction \mathit{acts}(M_i\|C)) \tag{7.49}$$

can be rewritten into

$$\left(\bar{p}_{\mathit{head}(\Theta_{i,j})}^{\mathit{shr}^{-1}(q)} \sum_{q'\in\Theta_{j,k}} p_{\mathit{head}(q')q'}\right) \sum_{q'\in\Theta_{j,k}} \frac{p_{\mathit{head}(q')q'}}{\sum_{q'\in\Theta_{j,k}} p_{\mathit{head}(q')q'}} P_{q'}^{H'}[a_j]\mathit{shr}(tr_{q'}^{H'} \restriction \mathit{acts}(M_i\|C)) \tag{7.50}$$

where (7.50) is obtained from (7.49) by expressing each $\bar{p}_{q'}^{\mathit{shr}^{-1}(q)}$ as $\bar{p}_{\mathit{head}(q')}^{\mathit{shr}^{-1}(q)} p_{\mathit{head}(q')q'}$, by grouping $\bar{p}_{\mathit{head}(\Theta_{i,j})}^{\mathit{shr}^{-1}(q)}$, which is equal to $\bar{p}_{\mathit{head}(q')}^{\mathit{shr}^{-1}(q)}$ for each q' os $\Theta_{i,j}$, by substituting $P_{q'}^{H'}[a_j]$ for $P_{q'}^{H'}[\mathit{acts}(M_i\|C]$ (action a_j is the only action of $M_i\|C$ that can be performed from q' due to the structure of H'), and by multiplying and dividing by $\sum_{q'\in\Theta_{j,k}} p_{\mathit{head}(q')q'}$.

Observe that each transition that appears in (7.50) is generated by some transitions of $M_i\|C$. Thus, the transition of (7.50) is generated by a combined transition of $M_i\|C$. Denote this transition by $tr_{j,k}$. Then, in Expression (7.47) it is possible to substitute each subexpression $\sum_{q'\in\Theta_{j,k}} \bar{p}_{q'}^{\mathit{shr}^{-1}(q)} P_{q'}^{H'}[\mathit{acts}(M_i\|C)]\mathit{shr}(tr_{q'} \restriction \mathit{acts}(M_i\|C))$ with $(\bar{p}_{\mathit{head}(q')}^{\mathit{shr}^{-1}(q)} \sum_{q'\in\Theta_{j,k}} p_{\mathit{head}(q')q'})tr_{j,k}$. This is enough to conclude.

b. For each state q of H,

$$P_H[C_q] = \sum_{q' \in min(shr^{-1}(q))} P_{H'}[C_{q'}]. \tag{7.51}$$

This is shown by induction on the length of q. If q consists of a start state only, then the result is trivial. Otherwise, from the definition of the probability of a cone and (7.47),

$$P_H[C_{qas}] = \sum_{q' \in shr^{-1}(q)} P_{H'}[C_{q'}] P_{q'}^{H'}[a \times shr^{-1}(qas)]. \tag{7.52}$$

Observe that the states of $min(shr^{-1}(qas))$ are the states that appear in $(a \times shr^{-1}(qas)) \cap \Omega_{q'}$ for some $q' \in shr^{-1}(q)$. Thus, $P_H[C_{qas}] = \sum_{q' \in min(shr^{-1}(qas))} P_{H'}[C_{q'}]$.

c. $tdistr(H) = tdistr(H') \upharpoonright acts(M_i \| C)$.

Let β be a finite trace of H or the projection of a finite trace of H'. Then $\{\alpha \in \Omega_{H'} \mid \beta \leq trace(\alpha) \upharpoonright acts(M_i \| C)\}$ can be expressed as a union of disjoint cones $\cup_{q \in \Theta} C_q$ where

$$\Theta = \{q \in states(H') \mid trace(q) \upharpoonright acts(M_i \| C) = \beta, lact(q) = lact(\beta)\}. \tag{7.53}$$

Observe that Θ is a set of closed states. The set $shr(\Theta)$ is the set

$$shr(\Theta) = \{q \in states(H) \mid trace(q) = \beta, lact(q) = lact(\beta)\}, \tag{7.54}$$

which is a characterization of $\{\alpha \in \Omega_H \mid \beta \leq trace(\alpha)\}$ as a union of disjoint cones. Observe that $min(shr^{-1}(shr(\Theta))) = \Theta$. Moreover, for each $q_1 \neq q_2$ of $shr(\Theta)$, $shr^{-1}(q_1) \cap shr^{-1}(q_2) = \emptyset$. Thus, from (7.51), $P_{H'}[\cup_{q \in \Theta} C_q] = P_H[q \in shr(\Theta) C_q]$.

To complete the proof of (7.43), it is enough to observe that $H_1 = shr(H_1')$. Property (7.43) is then expressed by property (c).

Suppose by contradiction that it is possible to obtain $\mathcal{D}'$ from $M_2 \| C'$. Consider the scheduler that leads to $\mathcal{D}'$ in $M_2 \| C'$, and let H_2' be the corresponding probabilistic execution. First, we build a new probabilistic execution H_2'' of $M_2 \| C'$ whose trace distribution is $\mathcal{D}'$, such that there is no action of M_2 between each state of the kind $\bullet_i$ and the occurrence of the corresponding external action of C, and such that all the transitions between a state of the kind $\bullet_j$ and the corresponding occurrences of action a_j are scheduled. Then we let $H_2 = shr(H_2'')$. This leads to a contradiction since $tdistr(H_2) = \mathcal{D}$. The rest of the proof is dedicated to the construction of H_2''.

For each state q of H_2', let $shf(q)$ be the set of sequences q' that can be obtained from q as follows: each sequence

$$(s_0, \bullet_j) b_1 (s_1, \bullet) \cdots b_k (s_k, \bullet) a_j (s, c)$$

is replaced with

$$(s_0, \bullet_j) b_{i_1} (s_0, \bullet) \cdots b_{i_l} (s_0, \bullet) a_j (s_0, c) b_{k_1} (s_{k_1}, c) \cdots b_{k_m} (s, c)$$

where $i_1, \ldots, i_l$ is the ordered sequence of the indexes of the b's that are actions of C', and $k_1, \ldots, k_m$ is the ordered sequence of the indexes of the b's that are actions of M_2; each sequence

$$(s_0, \bullet_j) b_1 (s_1, \bullet) \cdots b_k (s_k, \bullet)$$

occurring at the end of q either is replaced with

$$(s_0, \bullet_j) b_{i_1} (s_0, \bullet) \cdots b_{i_l} (s_0, \bullet) \frown \alpha \frown (s_0, \bullet) a_j (s_0, c) b_{k_1} (s_{k_1}, c) \cdots b_{k_m} (s, c)$$

where $i_1, \ldots, i_l$ is the ordered sequence of the indexes of the b's that are actions of C', $k_1, \ldots, k_m$ is the ordered sequence of the indexes of the b's that are actions of M_2, and α, called an *extension* for q, is an arbitrary execution fragment of $M_2 \| C'$ that leads to the occurrence of a_j, or, is replaced with a prefix of $(s_0, \bullet_j) b_{i_1} (s_0, \bullet) \cdots b_{i_l} (s_0, \bullet)$. Then,

$$\mathit{states}(H_2'') \triangleq \bigcup_{q \in \mathit{states}(H_2')} \mathit{shf}(q). \tag{7.55}$$

Let $(q, \mathcal{P})$ be a restricted transition of H_2', and suppose that only actions of M_2 and V_{start} occur. Let q' be a state of $\mathit{shf}(q)$. Then, for each $(a, q_1) \in \Omega$ there is exactly one $q_1' \in \mathit{shf}(q_1)$ such that $q' \leq q_1'$ and $|q_1'| = |q'| + 1$. Denote such q_1' by $\mathit{shf}_{q'}(q_1)$. Let $\Omega' = \{(a, \mathit{shf}_{q'}(q_1) \mid (a, q_1) \in \Omega\}$, and let, for each $(a, q_1') \in \Omega'$, $P'[(a, q_1')] = P[(a \times \mathit{shf}^{-1}(q_1'))]$, where $\mathit{shf}^{-1}(q)$ is the set of states q' of H_2' such that $q \in \mathit{shf}(q')$. Then define the transition $\mathit{shf}_{q'}((q, \mathcal{P}))$ to be

$$\mathit{shf}_{q'}((q, \mathcal{P})) \triangleq (q', \mathcal{P}). \tag{7.56}$$

For each state q of H_2'', let $\mathit{min}(\mathit{shf}^{-1}(q))$ be the set of minimal states of $\mathit{shf}^{-1}(q)$ under prefix ordering. Let q be a closed state of H_2'', and let $q' \in \mathit{shf}^{-1}(q)$. If q' is an open state, then let α be the extension for q' that is used in q, and let $E_{q'}^q$ be the product of the probabilities of the edges of α. For each state q' of $\mathit{shf}^{-1}(q)$, where q is closed, let

- $p_{q'}^q \triangleq P_{H_2'}[C_{q'}]$ if q' is closed;
- $p_{q'}^q \triangleq P_{H_2'}[C_{q'}] E_{q'}^q$ if q' is open.

For each $q' \in \mathit{shf}^{-1}(q)$, let

$$\bar{p}_{q'}^{\mathit{shf}^{-1}(q)} \triangleq \frac{p_{q'}^q}{\sum_{q'' \in \mathit{min}(\mathit{shf}^{-1}(q))} p_{q''}^q}. \tag{7.57}$$

If q is open, then the transition enabled from q in H_2'' is the one due to the transition of C' enabled from $\mathit{lstate}(q) \lceil C'$; if q is closed, then the transition enabled from q in H_2'' is

$$\sum_{q' \in \mathit{shf}^{-1}(q)} \bar{p}_{q'}^{\mathit{shf}^{-1}(q)} P_{q'}^{H_2'}[\mathit{acts}(H_2') \backslash (\mathit{acts}(C) \cup V_2)] \tag{7.58}$$
$$\mathit{shf}_q(\mathit{tr}_{q'}^{H_2'} \upharpoonright (\mathit{acts}(H_2') \backslash (\mathit{acts}(C) \cup V_2))).$$

The probabilistic execution H_2'' satisfies the following properties.

a. H_2'' is a probabilistic execution of $M_2\|C'$.

The fact that each state of H_2'' is reachable can be shown by a simple inductive argument; the fact that each state of H_2'' is a finite execution fragment of $M_2\|C'$ follows from a simple analysis of the definition of *shf*.

We need to check that for each state q of H_2'' the transition enabled from q in H_2'' is generated by a combination of transitions of $M_2\|C'$. If q is an open state, then the result follows immediately from the definition of the transition relation of H_2''. If q is a closed state, then consider a state $q' \in shf^{-1}(q)$. The transition $tr_{q'}^{H_2'} \restriction (acts(H_2')\backslash V_2)$, which appears in Expression (7.58), can be expressed as $\sum_i p_i(q' \frown tr_i)$, where each tr_i is a transition of $M_2\|C'$ enabled from $lstate(q')$. We distinguish two cases.

1. tr_i is a transition of M_2.
   Then $tr_i = ((s,c), a, \mathcal{P} \otimes \mathcal{D}(c))$ for some action a and probability space $\mathcal{P}$, where $(s,c) = lstate(q')$. Let $lstate(q) = (s', c')$. Then, $s' = s$. Define tr_i' to be the transition $((s,c'), a, \mathcal{P} \otimes \mathcal{D}(c'))$. Then tr_i' is a transition of $M_2\|C'$ and $shf_q(q' \frown tr_i) = q \frown tr_i'$.
2. tr_i is a transition of C'.
   Then $tr_i = ((s,c), a, \mathcal{D}(s) \otimes \mathcal{P})$ for some action a and probability space $\mathcal{P}$, where $(s,c) = lstate(q')$. Let $lstate(q) = (s', c')$. Then, $s' = s$ and $c = c'$ (q is closed). Define tr_i' to be tr_i. Then tr_i' is a transition of $M_2\|C'$ and $shf_q(q' \frown tr_i) = q \frown tr_i'$.

Observe that *shf* distributes over combination of transitions, and thus, the transition $shf_q(t_{q'}^{H_2'} \restriction (acts(H_2')\backslash V_2))$ can be expressed as $\sum_i p_i(q \frown t_i')$, which is generated by a combination of transitions of $M_2\|C'$.

b. For each state q of H_2'',

$$P_{H_2''}[C_q] = \begin{cases} \sum_{q' \in min(shf^{-1}(q))} p_{q'}^{q} & \text{if } q \text{ is closed,} \\ \sum_{q' \in min(shf^{-1}(q))} P_{H_2'}[C_{q'}] & \text{if } q \text{ is open.} \end{cases} \tag{7.59}$$

The proof is by induction on the length of q. If q consists of a start state only, then the result is trivial. Otherwise, consider $P_{H_2''}[C_{qas}]$. We distinguish two cases.

1. q is open.
   In this case a is an action of $V_2 \cup acts(C)$, and each state of $shf^{-1}(q)$ is open. From the definition of the probability of a cone and induction,

$$P_{H_2''}[C_{qas}] = \left(\sum_{q' \in min(shf^{-1}(q))} P_{H_2'}[C_{q'}] \right) P_q^{H_2''}[(a, qas)]. \tag{7.60}$$

   We distinguish two other cases.

   (a) $a \in V_2$.
   Observe that all the states of $min(shf^{-1}(q))$ enable the same transition of C' that is enabled from q. Moreover, for each $q' \in min(shf^{-1}(q))$, action a occurs with probability 1 (in $\mathcal{D}'$ each occurrence of a start action is followed by an

external action with probability 1), and the probability of reaching a state of $min(shf^{-1}(qas))$ given that a occurs is $P_q^{H_2''}[(a, qas)]$ (recall that q enables only action a). Since all the states of $min(shf^{-1}(qas))$ are open and have a prefix in $min(shf^{-1}(q))$, we can conclude

$$P_{H_2''}[C_{qas}] = \sum_{q' \in min(shf^{-1}(qas))} P_{H_2'}[C_{q'}]. \tag{7.61}$$

(b) $a \in acts(C)$.

From the definition of H_2'', $P_q^{H_2''}[(a, qas)] = 1$. Observe that all the states of $min(shf^{-1}(q))$ enable the same transition of C that is enabled from q. Moreover, for each $q' \in min(shf^{-1}(q))$, action a occurs with probability 1 (in $\mathcal{D}'$ each occurrence of a start action is followed by an external action with probability 1), leading to a state of $shf^{-1}(qas)$ for sure (recall that q enables only action a). Thus, for each $q' \in shf^{-1}(q)$,

$$P_{H_2'}[C_{q'}] = \sum_{q'' \in min(shf^{-1}(qas)) | q' \leq q''} P_{H_2'}[C_{q''}]. \tag{7.62}$$

Combining (7.60) and (7.62), we obtain

$$P_{H_2''}[C_{qas}] = \sum_{q' \in min(shf^{-1}(qas))} P_{H_2'}[C_{q'}]. \tag{7.63}$$

For each $q' \in min(shf^{-1}(qas))$, if q' is open, then $p_{q'}^{qas} = P_{H_2'}[C_{q'}]$ by definition; if q' is closed, then $p_{q'}^{qas} = P_{H_2'}[C_{q'}]$ since $E_{q'}^{qas} = 1$ (no α must be added by shf to get q' from qas). Thus, (7.63) becomes

$$P_{H_2''}[C_{qas}] = \sum_{q' \in min(shf^{-1}(qas))} p_{q'}^{qas}. \tag{7.64}$$

2. q is closed.

In this case, from the definition of the probability of a cone and (7.58),

$$P_{H_2''}[C_{qas}] = P_{H_2''}[C_q] \left(\sum_{q' \in shf^{-1}(q)} \bar{p}_{q'}^{shf^{-1}(q)} P_{q'}^{H_2'}[a \times shf^{-1}(qas)] \right) \tag{7.65}$$

From induction, the definition of $\bar{p}_{q'}^{shf^{-1}(q)}$, and an algebraic simplification,

$$\begin{aligned} P_{H_2''}[C_{qas}] \quad = \quad & \sum_{q' \in shf^{-1}(q) | closed(q')} P_{H_2'}[C_{q'}] P_{q'}^{H_2'}[a \times shf^{-1}(qas)] + \\ & \sum_{q' \in shf^{-1}(q) | open(q')} P_{H_2'}[C_{q'}] E_{q'}^{q} P_{q'}^{H_2'}[a \times shf^{-1}(qas)]. \end{aligned} \tag{7.66}$$

We distinguish two subcases.

(a) qas is open.

In this case each state q' of $shf^{-1}(q)$ such that $P_{q'}^{H_2'}[a \times shf^{-1}(qas)] > 0$ is closed, and thus only the first summand of (7.66) is used. Moreover, for each q' of $shf^{-1}(q)$ the set $\Omega_{q'}^{H_2'} \cap a \times shf^{-1}(qas)$ is made of open states $q'as'$ such that $E_{q'as'}^{qas} = 1$. Observe that all the states of $min(shf^{-1}(qas))$ are captured. Thus,

$$P_{H_2''}[C_{qas}] = \sum_{q' \in min(shf^{-1}(qas))} p_{q'}^{q}. \tag{7.67}$$

(b) qas is closed.
In this case, for each $q' \in shf^{-1}(q)$, if q' is closed, then all the states reached in $\Omega_{q'} \cap (\{a\} \times shf^{-1}(qas))$ are closed, and if q' is open, then all the states reached in $\Omega_{q'} \cap (\{a\} \times shf^{-1}(qas))$ are open and the extension α does not change, i.e., the term E does not change. Observe that all the states of $min(shf^{-1}(qas))$ are captured. Thus,

$$P_{H''_2}[C_{qas}] = \sum_{q' \in min(shf^{-1}(qas))} p^q_{q'}. \tag{7.68}$$

c. $tdistr(H'_2) = tdistr(H''_2)$.

Let β be a finite trace of H'_2 or H''_2. Then $\{\alpha \in \Omega_{H'_2} \mid \beta \leq trace(\alpha)\}$ can be expressed as a union of disjoint cones $\cup_{q \in \Theta} C_q$. We distinguish two cases.

1. β does not end with an action of C.
Then

$$\Theta = \{q \in states(H') \mid trace(q) = \beta, lact(q) = lact(\beta)\}. \tag{7.69}$$

The set $\Theta' = \{q \in shf(\Theta) \mid lact(q) = lact(\beta)\}$ is a characterization of $\{\alpha \in \Omega_{H''_2} \mid \beta \leq trace(\alpha)\}$ as a union of disjoint cones. Observe that $min(shf^{-1}(\Theta')) = \Theta$ and that for each $q_1 \neq q_2$ of Θ', $min(shf^{-1}(q_1)) \cap min(shf^{-1}(q_2)) = \emptyset$. Thus, from (7.51), $P_{H'_2}[\{\alpha \in \Omega_{H'_2} \mid \beta \leq trace(\alpha)\}] = P_{H''_2}[\{\alpha \in \Omega_{H''_2} \mid \beta \leq trace(\alpha)\}]$.

2. β ends with an action of C.
In this case $\beta = \beta' a_j$ for some action $a_j \in acts(C)$. Since in H'_2 and H''_2 after the occurrence of a state $\bullet_j$ the corresponding action a_j occurs with probability 1, we can assume that all the states of Θ end in $\bullet_j$, i.e.,

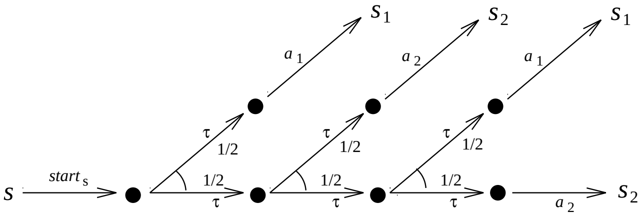
$$\Theta = \{q \in states(H') \mid trace(q) = \beta', \text{ and } lstate(q) \text{ is one of the } \bullet_j\text{'s}\}. \tag{7.70}$$

Then the set $\Theta' = min(shf(\Theta))$ is a characterization of $\{\alpha \in \Omega_{H''_2} \mid \beta \leq trace(\alpha)\}$ as a union of disjoint cones. Observe that all the elements of Θ are open. Property (7.59) is sufficient to conclude. ■

Lemma 7.5.7 *Let C be a distinguishing binary separated context for two probabilistic automata M_1 and M_2. Then there exists a distinguishing total binary separated context C' for M_1 and M_2 where all the probabilistic transitions have a uniform distribution. C' is called a* balanced separated context*.*

Proof. We achieve the result in two steps. First we decompose a binary probabilistic transition into several binary uniform probabilistic transitions, leading to a new distinguishing context C_1; then we use Lemma 7.5.4 to make C_1 into a cycle-free context.

The context C_1 is obtained from C by expressing each probabilistic transition of C by means of, possibly infinitely many, binary probabilistic transitions. For each state s of C, let $start_s$ be a new action. If s enables a probabilistic transition with actions a_1, a_2 to states s_1, s_2, respectively, and with probabilities p_1, p_2, respectively, then C_1 enables from s a deterministic transition with action $start_s$. Then, C_1 enables an internal probabilistic transition with a uniform distribution. If $p_1 > p_2$ ($p_2 > p_1$), then one of the states that is reached enables a deterministic transition with action a_1 (a_2). The other state enables a new internal probabilistic transition with a uniform binary distribution, and the transitions from the successive states are determined by giving a_1 probability $2(p_1 - 1/2)$ and a_2 probability $2p_2$ (a_1 probability $2p_1$ and a_2 probability $2(p_2 - 1/2)$). If $p_1 = p_2$, then one state enables a_1, and the other state enables a_2. For example, if $p_1 = 5/8$ and $p_2 = 3/8$, then the corresponding transitions of C_1 are represented below. Let $\mathcal{D}$ be a trace distribution of $M_1 \| C$ that is not a trace distribution of $M_2 \| C$. Consider a probabilistic execution H_1 of $M_1 \| C$ whose trace distribution is $\mathcal{D}$, and consider the scheduler that leads to H_1 in $M_1 \| C$. Apply to $M_1 \| C_1$ the same scheduler with the following modification: whenever a probabilistic transition of C is scheduled, schedule the *start* action from C_1, then schedule the internal transitions to resolve the probabilistic choice, and finally schedule the chosen action. Denote the resulting probabilistic execution by H_1' and the resulting trace distribution by $\mathcal{D}'$. Then,



$$\mathcal{D}' \upharpoonright acts(M_1 \| C) = \mathcal{D}. \tag{7.71}$$

To prove (7.71), we define a new construction shr_1, similar to *shr*, to be applied to probabilistic executions of $M_i \| C_1$ such that no action of M_i occurs between the occurrence of a $start_s$ action and the occurrence of one of the corresponding external actions of C, and such that all the transitions of C_1 between the occurrence of an action $start_s$ and the occurrence of one of the corresponding external actions of C are scheduled. The new function is identical to *shr* if we consider each state reached immediately after the occurrence of a start action like the states $\bullet_j$ used in Lemma 7.5.6. We leave the details to the reader.

Suppose by contradiction that it is possible to obtain $\mathcal{D}'$ from $M_2 \| C_1$. Consider the scheduler that leads to $\mathcal{D}'$ in $M_2 \| C_1$, and let H_2' be the corresponding probabilistic execution. First, we build a new probabilistic execution H_2'' of $M_2 \| C_1$ whose trace distribution is $\mathcal{D}'$, such that no action of M_i occurs between the occurrence of a $start_s$ action and the occurrence of one of the corresponding external action of C, and such that all the transitions of C_1 between the occurrence of an action $start_s$ and the occurrence of one of the corresponding external action of C are scheduled. Then we let $H_2 = shr_1(H_2'')$. This leads to a contradiction since $tdistr(H_2) = \mathcal{D}$.

The construction of H_2'', which is left to the reader, is the same as *shf* if we consider each state reached immediately after the occurrence of a start action like the states $\bullet_j$ used in Lemma 7.5.6. ■

Lemma 7.5.8 *Let C be a distinguishing balanced separated context for two probabilistic automata M_1 and M_2. Then there exists a distinguishing binary separated context C' for M_1 and M_2 with no internal actions and such that each action appears exactly in one edge of the transition tree. C' is called a* total balanced separated context.

Proof. The context C' is obtained from C by renaming all of its actions so that each edge of the new transition relation has its own action.

Let $\mathcal{D}$ be a trace distribution of $M_1\|C$ that is not a trace distribution of $M_2\|C$. Consider a probabilistic execution H_1 of $M_1\|C$ whose trace distribution is $\mathcal{D}$, and consider the scheduler that leads to H_1 in $M_1\|C$. Apply to $M_1\|C'$ the same scheduler with the following modification: whenever a transition of C is scheduled, schedule the corresponding transition of C'. Denote the resulting probabilistic execution by H_1' and the corresponding trace distribution by $\mathcal{D}'$. From construction, H_1 and H_1' are the same up to the names of the actions of C. Thus, if ρ' is the function that maps each action of C' to its original name in C, $\mathcal{D} = \rho'(\mathcal{D}')$ (the renaming of a trace distribution is the probability space induced by the function that renames traces).

Suppose by contradiction that it is possible to obtain $\mathcal{D}'$ from $M_2\|C'$. Consider the scheduler that leads to $\mathcal{D}'$ in $M_2\|C'$, and let H_2' be the corresponding probabilistic execution. Apply to $M_2\|C$ the same scheduler with the following modifications: whenever a transition of C' is scheduled, schedule the corresponding transition of C with the unrenamed actions. Let H_2 be the resulting probabilistic execution. From the construction, H_2 and H_2' are the same up to the names of the actions of C. Thus, $\mathit{tdistr}(H_2) = \rho'(\mathcal{D}') = \mathcal{D}$, which is a contradiction. ■

Lemma 7.5.9 *Let C be a distinguishing total balanced separated context for two probabilistic automata M_1 and M_2. Then there exists a distinguishing total balanced separated context C' for M_1 and M_2 that from every state enables two deterministic transitions and a probabilistic transition with a uniform distribution over two choices. C' is called a* complete context.

Proof. In this case it is enough to complete C by adding all the missing transitions and states. If $\mathcal{D}$ is a trace distribution of $M_1\|C$ that is not a trace distribution of $M_2\|C$, then it is enough to use on $M_1\|C'$ the same scheduler that is used in $M_1\|C$. In fact, since each new transition of C' has a distinct action, none of the new transitions of C' can be used in $M_2\|C'$ to generate $\mathcal{D}$. ■

Lemma 7.5.10 *Let C be a distinguishing complete context for two probabilistic automata M_1 and M_2. Then the principal context C_P is a distinguishing context for M_1 and M_2.*

Proof. The result is achieved in two steps. First the actions of C are renamed so that each state enables two deterministic transitions with actions *left* and *right*, respectively, and a probabilistic transition with actions *pleft* and *pright*. Call this context C_1. Then, by observing that each state s of C_1 is uniquely determined by the trace of the unique execution of C_1 that leads to s, all the states of C_1 are collapsed into a unique one.

Thus, we need to show only that C_1 is a distinguishing context. Let $\mathcal{D}$ be a trace distribution of $M_1\|C$ that is not a trace distribution of $M_2\|C$. Consider the scheduler that leads to $\mathcal{D}$ in $M_1\|C$, and apply to $M_1\|C_1$ the same scheduler with the following modification: whenever a transition of C is scheduled, schedule the corresponding transition of C_1. Denote the resulting trace distribution by $\mathcal{D}'$. Note that if we rename all the actions of C_1 into their original name in C, then we obtain $\mathcal{D}$.

Suppose by contradiction that it is possible to obtain $\mathcal{D}'$ from $M_2\|C_1$. Consider the scheduler that leads to $\mathcal{D}'$ in $M_2\|C_1$, and apply to $M_2\|C$ the same scheduler with the following modification: whenever a transition of C_1 is scheduled, schedule the corresponding transition of C. The resulting trace distribution is $\mathcal{D}$, which is a contradiction. ■

Lemma 7.5.11 *Let C_P be a distinguishing context for two probabilistic automata M_1 and M_2. Then the simple principal context, denoted by C, is a distinguishing context for M_1 and M_2.*

Proof. Let $\mathcal{D}$ be a trace distribution of $M_1 \| C_P$ that is not a trace distribution of $M_2 \| C_P$. Consider a probabilistic execution H_1 of $M_1 \| C_P$ whose trace distribution is $\mathcal{D}$, and consider the scheduler that leads to H_1 in $M_1 \| C_P$. Apply to $M_1 \| C$ the same scheduler with the following modification: whenever the probabilistic transition of C_P is scheduled, schedule the *start* action of C followed by the next transition of C that becomes enabled. Denote the resulting probabilistic execution by H_1' and the resulting trace distribution by $\mathcal{D}'$. Then,

$$\mathcal{D}' \upharpoonright acts(M_1 \| C_P) = \mathcal{D}. \tag{7.72}$$

To prove (7.72), we define a new construction shr_2, similar to shr, to be applied to probabilistic executions of $M_i \| C$ such that no action of M_i occurs between the occurrence of a *start* action and the occurrence of one of the actions *pleft* and *pright*, and such that the transitions labeled with *pleft* and *pright* occur whenever they are enabled. The new function is identical to shr if we consider each state reached after an action *start* as a state of the kind $\bullet_j$. We leave the details to the reader.

Suppose by contradiction that it is possible to obtain $\mathcal{D}'$ from $M_2 \| C$. Consider the scheduler that leads to $\mathcal{D}'$ in $M_2 \| C$, and let H_2' be the corresponding probabilistic execution. First, we build a new probabilistic execution H_2'' of $M_2 \| C$ whose trace distribution is $\mathcal{D}'$, such that no action of M_2 occurs between the occurrence of a *start* action and the occurrence of one of the actions *pleft* and *pright*, and such that the transitions labeled with *pleft* and *pright* occur whenever they are enabled. Then we let $H_2 = clp_2(H_2'')$. This leads to a contradiction since $tdistr(H_2) = \mathcal{D}$.

The construction of H_2'', which is left to the reader, is the same as shf if we consider each state reached immediately after the occurrence of a start action like the states $\bullet_j$ used in Lemma 7.5.6. ■

Proof of Theorem 7.5.1. Let $M_1 \sqsubseteq_{DC} M_2$. Then, from Lemma 7.5.11, $M_1 \| C_P \sqsubseteq_D M_2 \| C_P$. Conversely, let $M_1 \| C_P \sqsubseteq_D M_2 \| C_P$. Then, from Lemmas 7.5.3, 7.5.4, 7.5.5, 7.5.6, 7.5.7, 7.5.8, 7.5.9, and 7.5.10, $M_1 \sqsubseteq_{DC} M_2$. ■

7.6 Discussion

A trace-based semantics similar to ours is studied for generative processes by Jou and Smolka [JS90]. One of the processes of Jou and Smolka is essentially one of our probabilistic executions. The semantics of a process is given by a function, called a trace function, that associates a probability with each finite trace. Since our trace distributions are determined by the probabilities of the cones, our trace distributions are characterized completely by the trace functions of Jou and Smolka. In other words, the trace semantics of Jou and Smolka is the semantics that we use to say that two probabilistic executions have the same trace distribution.

Jou and Smolka define also a notion of a maximal trace function. Given a probabilistic execution H, the interpretation of a maximal trace function in our framework is a function that associates with each finite trace β the probability of the extended executions on Ω_H that end in δ and whose trace is β. Jou and Smolka show that the trace function of a process is sufficient to determine the maximal trace function of the process. In our trace distributions the maximal trace function of a probabilistic execution is given by the probability of each finite trace in the corresponding trace distribution. From the definition of a trace distribution the probability of each finite trace is determined uniquely by the probabilities of the cones, and thus the result of Jou and Smolka holds also in our framework.

Chapter 8

Hierarchical Verification: Simulations

8.1 Introduction

In Chapter 7 we have studied the trace distribution precongruence as an instance of the hierarchical method for the verification of probabilistic systems. Another instance of the hierarchical method is called the *simulation method*. According to the simulation method, rather than comparing two probabilistic automata through some abstract observations, two probabilistic automata are compared by establishing some relation between their states and by showing that the two probabilistic automata can simulate each other via the given relation. Standard work on simulation relations appears in [Mil89, Jon91, LV91]. Simulation relations are stronger than the trace preorder, and are often used as a sound proof technique for the trace preorder.

In this chapter we study how to extend some of the relations of [Mil89, Jon91, LV91] to the probabilistic framework. We start with the generalization of the simplest relations that do not abstract from internal computation, and we conclude with the generalization of the forward simulations of [LV91] that approximate closely the trace distribution preorder. We prove the equivalent of the Execution Correspondence Lemma [GSSL94] for probabilistic automata, which states that there is a strong connection between the probabilistic executions of two probabilistic automata related by some simulation relation. Finally, we use the new Execution Correspondence Lemma to prove that the existence of a probabilistic forward simulation is sufficient to prove the trace distribution precongruence relation.

8.2 Strong Simulations

One of the finest equivalence relations for ordinary automata would be graph isomorphism; however, it is widely recognized that graph isomorphism distinguishes too much. A coarser equivalence relation is strong bisimulation [Par81, Mil89], where two automata A_1 and A_2 are equivalent iff there is an equivalence relation between their states so that for each pair (s_1, s_2) of equivalent states,

if $s_1 \xrightarrow{a} s_1'$, then there exists a state s_2' equivalent to s_1' such that $s_2 \xrightarrow{a} s_2'$.

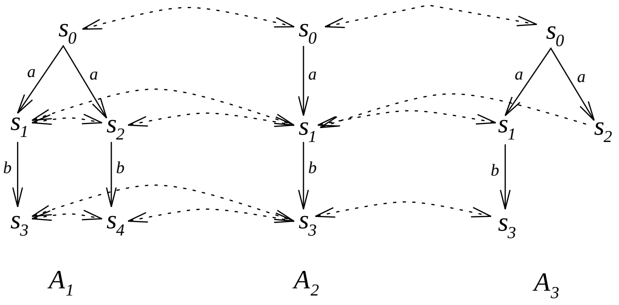


Figure 8-1: The difference between strong bisimulation and the kernel of strong simulation.

That is, A_1 and A_2 simulate each other. A preorder relation that is closely connected to strong bisimulation is strong simulation. An automaton A_1 is strongly simulated by another automaton A_2 iff there is a relation between the states of A_1 and the states of A_2 so that for each pair (s_1, s_2) of related states,

if $s_1 \xrightarrow{a} s'_1$, then there exists a state s'_2 such that $s_2 \xrightarrow{a} s'_2$ and s'_1 is related to s'_2.

The kernel of strong simulation is an equivalence relation that is coarser than bisimulation.

Example 8.2.1 (Strong simulation and strong bisimulation) Figure 8-1 shows the difference between strong bisimulation and the kernel of strong simulation. The double-arrow dotted links represent a strong bisimulation between A_1 and A_2; thus, A_1 and A_2 are strongly bisimilar. There is also a strong simulation from A_2 to A_3, expressed by the dotted lines that have an arrow pointing to A_3, and a strong simulation from A_3 to A_2, expressed by the dotted lines that have an arrow pointing to A_2. Thus, A_2 and A_3 are equivalent according to the kernel of strong simulation. However, there is no bisimulation between A_2 and A_3 since state s_2 of A_3 must be related to state s_1 of A_2 in order for A_2 to be able to simulate the transition $s_0 \xrightarrow{a} s_2$ of A_3, but then it is not possible to simulate the transition $s_1 \xrightarrow{b} s_3$ of A_2 from s_2 in A_3. ■

The extension of strong bisimulation and strong simulation to the probabilistic framework presents a problem due to the fact that a probabilistic transition leads to a probability distribution over states rather than to a single state. Thus, a relation over states needs to be lifted to distributions over states. Here we borrow an idea from [JL91].

Let $\mathcal{R} \subseteq X \times Y$ be a relation between two sets X, Y, and let $\mathcal{P}_1$ and $\mathcal{P}_2$ be two probability spaces of *Probs*(X) and *Probs*(Y), respectively. Then $\mathcal{P}_1$ and $\mathcal{P}_2$ are in relation $\sqsubseteq_{\mathcal{R}}$, written $\mathcal{P}_1 \sqsubseteq_{\mathcal{R}} \mathcal{P}_2$, iff there exists a *weight function* $w : X \times Y \rightarrow [0,1]$ such that

1. for each $x \in X$, $\sum_{y \in Y} w(x,y) = P_1[x]$,
2. for each $y \in Y$, $\sum_{x \in X} w(x,y) = P_2[y]$,
3. for each $(x,y) \in X \times Y$, if $w(x,y) > 0$ then $x \mathcal{R} y$.

Example 8.2.2 (Lifting of one relation) The idea behind the definition of $\sqsubseteq_{\mathcal{R}}$ is that each state of Ω_1 must be represented by some states of Ω_2, and similarly, each state of Ω_2 must represent one or more states of Ω_1. Figure 8-2 gives an example of two probability spaces that

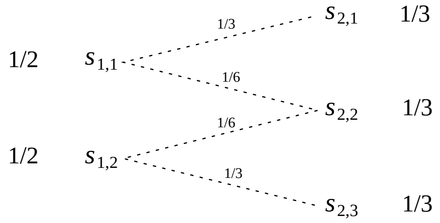


Figure 8-2: Lifting one relation.

are related. The dotted lines connect states that are related by $\mathcal{R}$. Thus, state $s_{1,1}$ can be represented by $s_{2,1}$ for a third of its probability, and by $s_{2,2}$ for the reminder. Similarly, state $s_{2,2}$ represents $s_{1,1}$ for one sixth of its probability and $s_{1,2}$ for the reminder. A useful property of $\sqsubseteq_{\mathcal{R}}$ is its preservation over combination of probability spaces. ■

If $\mathcal{R}$ is an equivalence relation, then we denote the relation $\sqsubseteq_{\mathcal{R}}$ alternatively by $\equiv_{\mathcal{R}}$. The reason for the alternative notation is that whenever $\mathcal{R}$ is an equivalence relation and $\mathcal{P}_1 \equiv_{\mathcal{R}} \mathcal{P}_2$, each equivalence class of $\mathcal{R}$ is assigned the same probability in $\mathcal{P}_1$ and $\mathcal{P}_2$ (cf. Lemma 8.2.2).

The definition of strong bisimulation and strong simulation for probabilistic automata are now straightforward. For convenience assume that M_1 and M_2 do not have common states. A *strong bisimulation* between two simple probabilistic automata M_1, M_2 is an equivalence relation $\mathcal{R}$ over $states(M_1) \cup states(M_2)$ such that

1. each start state of M_1 is related to at least one start state of M_2, and vice versa;
2. for each pair of states $s_1 \; \mathcal{R} \; s_2$ and each transition $s_1 \xrightarrow{a} \mathcal{P}_1$ of either M_1 or M_2, there exists a transition $s_2 \xrightarrow{a} \mathcal{P}_2$ of either M_1 or M_2 such that $\mathcal{P}_1 \equiv_{\mathcal{R}} \mathcal{P}_2$.

We write $M_1 \simeq M_2$ whenever $acts(M_1) = acts(M_2)$ and there is a strong bisimulation between M_1 and M_2.

A *strong simulation* between two simple probabilistic automata M_1, M_2 is a relation $\mathcal{R} \subseteq states(M_1) \times states(M_2)$ such that

1. each start state of M_1 is related to at least one start state of M_2;
2. for each pair of states $s_1 \; \mathcal{R} \; s_2$ and each transition $s_1 \xrightarrow{a} \mathcal{P}_1$ of M_1, there exists a transition $s_2 \xrightarrow{a} \mathcal{P}_2$ of M_2 such that $\mathcal{P}_1 \sqsubseteq_{\mathcal{R}} \mathcal{P}_2$.

We write $M_1 \sqsubseteq_{\mathrm{SS}} M_2$ whenever $acts(M_1) = acts(M_2)$ and there is a strong simulation from M_1 to M_2. We denote he kernel of strong simulation by $\equiv_{\mathrm{SS}}$. Because of Lemma 8.2.2, our strong bisimulations are the same as the bisimulations of [Han94], and our strong simulations are a generalization of the simulations of [JL91].

It is easy to check that $\simeq$ is an equivalence relation, that $\sqsubseteq_{\mathrm{SS}}$ is a preorder relation, and that both $\simeq$ and $\sqsubseteq_{\mathrm{SS}}$ are preserved by the parallel composition operator.

We conclude this section by proving two results about the lifting of a relation. The first result shows that the lifting of a relation is preserved by the combination of probability spaces; the second result shows that $\mathcal{P}_1 \equiv_{\mathcal{R}} \mathcal{P}_2$ iff $\mathcal{P}_1$ and $\mathcal{P}_2$ assign the same probability to each equivalence class of $\mathcal{R}$.

Lemma 8.2.1 *Let $\mathcal{P}_{X,i} \sqsubseteq_{\mathcal{R}} \mathcal{P}_{Y,i}$ via a weight function w_i, and let $\{p_i\}_{i\geq 0}$ be a family of real numbers between 0 and 1 such that $\sum_{i\geq 0} p_i = 1$. Then $\sum_{i\geq 0} p_i \mathcal{P}_{X,i} \sqsubseteq_{\mathcal{R}} \sum_{i\geq 0} p_i \mathcal{P}_{Y,i}$ via $\sum_{i\geq 0} p_i w_i$.*

Proof. Let $\mathcal{P}_X = \sum_{i\geq 0} p_i \mathcal{P}_{X,i}$, $\mathcal{P}_Y = \sum_{i\geq 0} p_i \mathcal{P}_{Y,i}$, and $w = \sum_{i\geq 0} p_i w_i$. Let $x \in \Omega_X$. Then $\sum_{y\in\Omega_Y} w(x,y) = \sum_{y\in\Omega_Y} \sum_{i\geq 0} p_i w_i(x,y) = \sum_{i\geq 0} p_i (\sum_{y\in\Omega_Y} w_i(x,y)) = \sum_{i\geq 0} p_i P_{X,i}[x] = P_X[x]$. Condition 2 of the definition of $\sqsubseteq_{\mathcal{R}}$ is verified similarly. For Condition 3, let $w(x,y) > 0$. Then there exists an i such that $w_i(x,y) > 0$, and thus $x \mathrel{\mathcal{R}} y$. ■

Lemma 8.2.2 *Let X, Y be two disjoint sets, $\mathcal{R}$ be an equivalence relation on $X \cup Y$, and let $\mathcal{P}_1$ and $\mathcal{P}_2$ be probability spaces of Probs(X) and Probs(Y), respectively. Then, $\mathcal{P}_1 \equiv_{\mathcal{R}} \mathcal{P}_2$ iff for each equivalence class C of $(X \cup Y)/\mathcal{R}$, $P_1[C \cap \Omega_1] = P_2[C \cap \Omega_2]$.*

Proof. Suppose that $\mathcal{P}_1 \equiv_{\mathcal{R}} \mathcal{P}_2$, and let w be the corresponding weight function. Then, for each equivalence class C of $(X \cup Y)/\mathcal{R}$,

$$P_1[C \cap \Omega_1] = \sum_{x\in C\cap\Omega_1} P_1[x] = \sum_{x\in C\cap\Omega_1} \sum_{y\in C\cap\Omega_2} w(x,y), \tag{8.1}$$

and

$$P_2[C \cap \Omega_2] = \sum_{y\in C\cap\Omega_2} P_2[y] = \sum_{y\in C\cap\Omega_2} \sum_{x\in C\cap\Omega_1} w(x,y). \tag{8.2}$$

From the commutativity and associativity of sum,

$$P_1[C \cap \Omega_1] = P_2[C \cap \Omega_2]. \tag{8.3}$$

Conversely, suppose that each equivalence class $(X \cup Y)/\mathcal{R}$ has the same probability in $\mathcal{P}_1$ and $\mathcal{P}_2$. We define $w(x,y)$ for each equivalence class of $(X \cup Y)/\mathcal{R}$, and we assume implicitly that w is 0 for all the pairs $(x,y) \in \Omega_1 \times \Omega_2$ that are not considered in the construction below. Consider any equivalence class C of $(X \cup Y)/\mathcal{R}$, and let $X' = C \cap \Omega_1$, and $Y' = C \cap \Omega_2$. From hypothesis we know that $P_1[X'] = P_2[Y']$. Let $x_1, x_2, \ldots$ be an enumeration of the points of X', and let $y_1, y_2, \ldots$ be an enumeration of the points of Y'. For each i, let $p_i = \sum_{k<i} P_1[x_i]$ and let $q_i = \sum_{k<i} P_2[y_i]$. Then

$$w(x_i, y_j) = \begin{cases} 0 & \text{if } p_{i+1} \leq q_j \text{ or } q_{j+1} \leq p_i \\ min(p_{i+1}, q_{j+1}) - max(p_i, q_j) & \text{otherwise.} \end{cases}$$

Informally, the construction above works as follows. Consider two intervals $[0, P_1[X']]$, and mark the first interval with the points p_i and the second interval with the points q_j. Each interval $[p_i, p_{i+1}]$ has length $P_1[x_i]$ and each interval $[q_j, q_{j+1}]$ has length $P_2[y_j]$. The weight function $w(x_i, y_j)$ is defined to be the length of the intersection of the intervals associated with x_i and y_j, respectively. It is simple to verify that w is a weight function for $\mathcal{P}_1$ and $\mathcal{P}_2$. ■

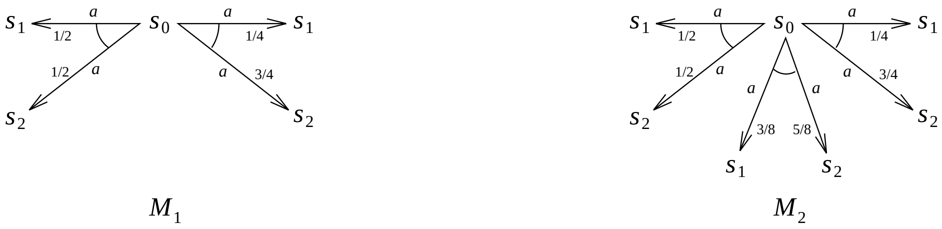


Figure 8-3: Combining transitions to simulate a transition.

8.3 Strong Probabilistic Simulations

In the definition of strong bisimulations and strong simulations we have not taken into account the fact that the nondeterminism can be resolved by combining several transitions probabilistically into a unique one. That is, a transition of a probabilistic automaton could be simulated by combining several transitions of another probabilistic automaton.

Example 8.3.1 (Combining transitions to simulate another transition) Consider the two probabilistic automata M_1 and M_2 of Figure 8-3. M_2 contains the transitions of M_1 plus a transitions that is obtained by combining probabilistically the transitions of M_1. For this reason there is no simulation from M_2 to M_1 (the additional transition cannot be simulated). On the other hand, M_1 and M_2 have exactly the same probabilistic executions, and therefore we do not see any reason to distinguish them. ■

Example 8.3.1 suggests two new relations, which are coarser than strong bisimulation and strong simulation, where the only difference is that a transition can be simulated by a probabilistic combination of transitions.

For convenience assume that M_1 and M_2 do not have common states. A *strong probabilistic bisimulation* between two simple probabilistic automata M_1, M_2 is an equivalence relation $\mathcal{R}$ over $states(M_1) \cup states(M_2)$ such that

1. each start state of M_1 is related to at least one start state of M_2, and vice versa;
2. for each pair of states $s_1 \mathcal{R} s_2$ and each transition $s_1 \xrightarrow{a} \mathcal{P}_1$ of either M_1 or M_2, there exists a combined transition $s_2 \xrightarrow{a}_{\mathrm{C}} \mathcal{P}_2$ of either M_1 or M_2 such that $\mathcal{P}_1 \equiv_{\mathcal{R}} \mathcal{P}_2$.

We write $M_1 \simeq_{\mathrm{P}} M_2$ whenever $acts(M_1) = acts(M_2)$ and there is a strong probabilistic bisimulation between M_1 and M_2.

A *strong probabilistic simulation* between two simple probabilistic automata M_1 and M_2 is a relation $\mathcal{R} \subseteq states(M_1) \times states(M_2)$ such that

1. each start state of M_1 is related to at least one start state of M_2;
2. for each pair of states $s_1 \mathcal{R} s_2$ and each transition $s_1 \xrightarrow{a} \mathcal{P}_1$ of M_1, there exists a combined transition $s_2 \xrightarrow{a}_{\mathrm{C}} \mathcal{P}_2$ of M_2 such that $\mathcal{P}_1 \sqsubseteq_{\mathcal{R}} \mathcal{P}_2$.

We write $M_1 \sqsubseteq_{\mathrm{SPS}} M_2$ whenever $acts(M_1) = acts(M_2)$ and there is a strong probabilistic simulation from M_1 to M_2. We denote the kernel of strong probabilistic simulation by $\equiv_{\mathrm{SPS}}$.

It is easy to check that $\simeq_{\mathrm{P}}$ is an equivalence relation, that $\sqsubseteq_{\mathrm{SPS}}$ is a preorder relation, and that both $\simeq_{\mathrm{P}}$ and $\sqsubseteq_{\mathrm{SPS}}$ are preserved by the parallel composition operator. It is easy as well to verify that a strong bisimulation is also a strong probabilistic bisimulation and that a strong simulation is also a strong probabilistic simulation.

8.4 Weak Probabilistic Simulations

The abstraction from internal computation can be obtained in the same way as for ordinary automata: a transition of a probabilistic automaton should be simulated by a collection of internal and external transitions of another probabilistic automaton. For the formal definition we use the weak combined transitions of Chapter 4.

For convenience assume that M_1 and M_2 do not have common states. A *weak probabilistic bisimulation* between two simple probabilistic automata M_1 and M_2 is an equivalence relation $\mathcal{R}$ over $states(M_1) \cup states(M_2)$ such that

1. each start state of M_1 is related to at least one start state of M_2, and vice versa;

2. for each pair of states $s_1 \mathcal{R} s_2$ and each transition $s_1 \xrightarrow{a} \mathcal{P}_1$ of either M_1 or M_2, there exists a weak combined transition $s_2 \overset{a\restriction ext(M_2)}{\Longrightarrow}_{\mathrm{C}} \mathcal{P}_2$ of either M_1 or M_2 such that $\mathcal{P}_1 \equiv_{\mathcal{R}} \mathcal{P}_2$.

We write $M_1 =_{\mathrm{P}} M_2$ whenever $ext(M_1) = ext(M_2)$ and there is a weak probabilistic bisimulation between M_1 and M_2.

A *weak probabilistic simulation* between two simple probabilistic automata M_1 and M_2 is a relation $\mathcal{R} \subseteq states(M_1) \times states(M_2)$ such that

1. each start state of M_1 is related to at least one start state of M_2;

2. for each pair of states $s_1 \mathcal{R} s_2$ and each transition $s_1 \xrightarrow{a} \mathcal{P}_1$ of M_1, there exists a weak combined transition $s_2 \overset{a\restriction ext(M_2)}{\Longrightarrow}_{\mathrm{C}} \mathcal{P}_2$ of M_2 such that $\mathcal{P}_1 \sqsubseteq_{\mathcal{R}} \mathcal{P}_2$.

We write $M_1 \sqsubseteq_{\mathrm{WPS}} M_2$ whenever $ext(M_1) = ext(M_2)$ and there is a weak probabilistic simulation from M_1 to M_2. We denote the kernel of weak probabilistic simulation by $\equiv_{\mathrm{WPS}}$.

It is easy to verify that a strong probabilistic bisimulation is also a weak probabilistic bisimulation and that a strong probabilistic simulation is also a weak probabilistic simulation. However, it is not as easy to verify that $=_{\mathrm{P}}$ is an equivalence relation, that $\sqsubseteq_{\mathrm{WPS}}$ is a preorder relation, and that both $=_{\mathrm{P}}$ and $\sqsubseteq_{\mathrm{WPS}}$ are preserved by the parallel composition operator. The verification of these properties is a simplification of the verification of the same properties for the relation of the next section. For this reason we omit the proofs from this section.

8.5 Probabilistic Forward Simulations

One of the main results of this chapter is that all the relations presented so far are sound for the trace distribution precongruence. However, none of the relations of the previous sections allow for one probabilistic operation to be implemented by several probabilistic operations.

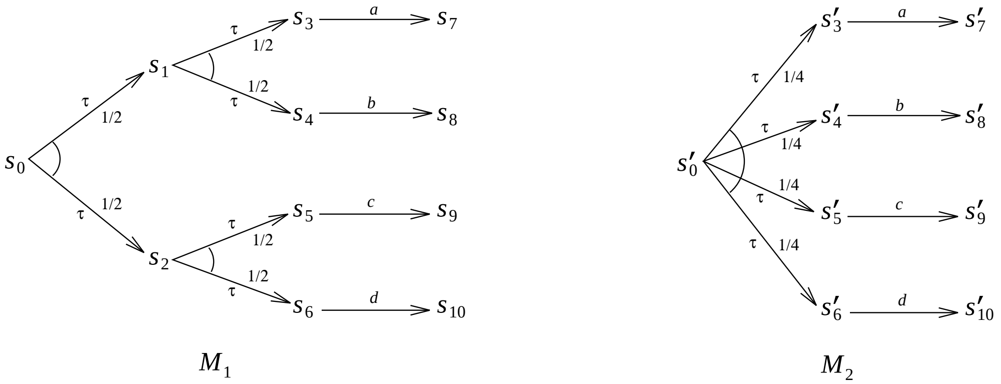


Figure 8-4: Implementation of a probabilistic transition with several probabilistic transitions.

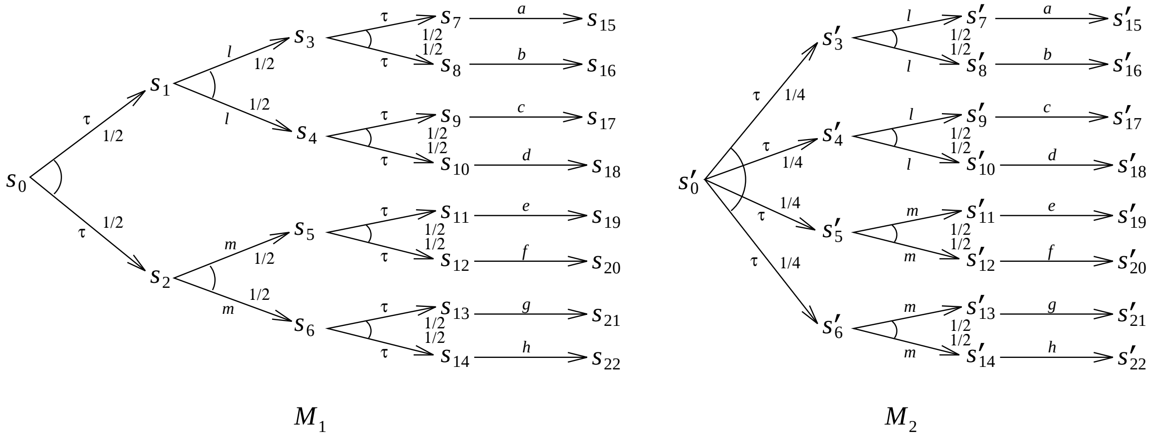


Figure 8-5: A more sophisticated implementation.

Example 8.5.1 (Weak probabilistic simulations are too coarse) Consider the two probabilistic automata of Figure 8-4. The probabilistic automaton M_2, which chooses internally one element out of four with probability $1/4$ each, is implemented by the probabilistic automaton M_1, which flips two fair coins to make the same choice. However, the first transition of M_1 cannot be simulated by M_2 since the probabilistic choice of M_2 is not resolved completely yet in M_1. This situation suggests a new preorder relation where a state of M_1 can be related to a probability distribution over states of M_2. The informal idea behind a relation $s_1 \mathcal{R} \mathcal{P}_2$ is that s_1 represents an intermediate stage of M_1 in reaching the distribution $\mathcal{P}_2$. For example, in Figure 8-4 state s_1 would be related to a uniform distribution $\mathcal{P}$ over states s'_3 and s'_4 ($\mathcal{P} = \mathcal{U}(s'_3, s'_4)$), meaning that s_1 is an intermediate stage of M_1 in reaching the distribution $\mathcal{P}$.

It is also possible to create examples where the relationship between s and $\mathcal{P}$ does not mean simply that s is an intermediate stage of M_1 in reaching the distribution $\mathcal{P}$, but rather that s is an intermediate stage in reaching a probability distribution that can be reached from $\mathcal{P}$. Consider the two probabilistic automata of Figure 8-5. Although not evident at the moment, M_1 and M_2 are in the trace distribution precongruence relation, i.e., $M_1 \sqsubseteq_{DC} M_2$. Following the same idea as for the example of Figure 8-4, state s_1 is related to $\mathcal{U}(s'_3, s'_4)$. However, s_1 is not an intermediate stage of M_1 in reaching $\mathcal{U}(s'_3, s'_4)$, since s_1 enables a transition labeled with an external action l, while in M_2 no external action occurs before reaching $\mathcal{U}(s'_3, s'_4)$. Rather, from s'_3 and s'_4 there are two transitions labeled with l, and thus the only way to simulate the transition $s_1 \xrightarrow{l} \mathcal{U}(s_3, s_4)$ from $\mathcal{U}(s'_3, s'_4)$ is to perform the two transitions labeled with l, which lead to the distribution $\mathcal{U}(s'_7, s'_8, s'_9, s'_{10})$. Now the question is the following: in what sense does $\mathcal{U}(s'_7, s'_8, s'_9, s'_{10})$ represent $\mathcal{U}(s_3, s_4)$? The first observation is that s_3 can be seen as an intermediate stage in reaching $\mathcal{U}(s'_7, s'_8)$, and that s_4 can be seen as an intermediate stage in reaching $\mathcal{U}(s'_9, s'_{10})$. Thus, s_3 is related to $\mathcal{U}(s'_7, s'_8)$ and s_4 is related to $\mathcal{U}(s'_9, s'_{10})$. The second observation is that $\mathcal{U}(s'_7, s'_8, s'_9, s'_{10})$ can be expressed as $1/2\mathcal{U}(s'_7, s'_8) + 1/2\mathcal{U}(s'_9, s'_{10})$. Thus, $\mathcal{U}(s'_7, s'_8, s'_9, s'_{10})$ can be seen as a combination of two probability spaces, each one representing an element of $\mathcal{U}(s_3, s_4)$. This recalls the lifting of a relation that we introduced at the beginning of this chapter. ■

Based on Example 8.5.1, we can move to the formal definition of a probabilistic forward simulation. A *probabilistic forward simulation* between two simple probabilistic automata M_1 and M_2 is a relation $\mathcal{R} \subseteq states(M_1) \times Probs(states(M_2))$ such that

1. each start state of M_1 is related to at least one Dirac distribution over a start state of M_2;
2. for each $s \; \mathcal{R} \; \mathcal{P}'$, if $s \xrightarrow{a} \mathcal{P}_1$, then
   (a) for each $s' \in \Omega'$ there exists a probability space $\mathcal{P}_{s'}$ such that $s' \stackrel{a\upharpoonright ext(M_2)}{\Longrightarrow_C} \mathcal{P}_{s'}$, and
   (b) there exists a probability space $\mathcal{P}'_2$ of $Probs(Probs(states(M_2)))$ satisfying $\mathcal{P}_1 \sqsubseteq_{\mathcal{R}} \mathcal{P}'_2$,

   such that $\sum_{s' \in \Omega'} P'[s']\mathcal{P}_{s'} = \sum_{\mathcal{P} \in \Omega'_2} P'_2[\mathcal{P}]\mathcal{P}$.

We write $M_1 \sqsubseteq_{FS} M_2$ whenever $ext(M_1) = ext(M_2)$ and there is a probabilistic forward simulation from M_1 to M_2.

Example 8.5.2 (A probabilistic forward simulation) The probabilistic forward simulation for the probabilistic automata M_1 and M_2 of Figure 8-5 is the following: s_0 is related to $\mathcal{U}(s'_0)$; each state s_i, $i \geq 7$, is related to $\mathcal{D}(s'_i)$; each state s_i, $1 \leq i \leq 6$, is related to $\mathcal{U}(s'_{2i+1}, s'_{2i+2})$. It is an easy exercise to check that this relation is a probabilistic forward simulation. Observe also that there is no probabilistic forward simulation from M_2 to M_1. Informally, s'_3 cannot be simulated by M_1, since the only candidate state to be related to s'_1 is s_1, and s_1 does not contain all the information contained in s'_3. The formal way to see that there is no probabilistic forward simulation from M_2 to M_1 is to observe that M_2 and M_1 are not in the trace distribution precongruence relation and then use the fact that probabilistic forward simulations are sound for the trace distribution precongruence relation (cf. Section 8.7). In $M_2 \| C_P$ it is possible force action *left* to be scheduled exactly when M_2 is in s'_3, and thus it is possible to create a correlation between action *left* and actions a and b; in $M_1 \| C_P$ such a correlation cannot be created since action *left* must be scheduled before action l. ■

It is easy to check that a weak probabilistic simulation is a special case of a probabilistic forward simulation where each state of M_1 is related to a Dirac distribution. The verification that $\sqsubseteq_{FS}$

is a preorder that is preserved by parallel composition is more complicated. In this section we show that $\sqsubseteq_{FS}$ is preserved by parallel composition; the proof that $\sqsubseteq_{FS}$ is a preorder is postponed to Section 8.6.4.

Proposition 8.5.1 *$\sqsubseteq_{FS}$ is preserved by the parallel composition operator.*

Proof. Let $M_1 \sqsubseteq_{FS} M_2$, and let $\mathcal{R}$ be a probabilistic forward simulation from M_1 to M_2. Let $\mathcal{R}'$ be a relation between $states(M_1) \times states(M_3)$ and $Probs(states(M_2) \times states(M_3))$, defined as follows:

$$(s_1, s_3)\ \mathcal{R}'\ \mathcal{P} \text{ iff } \mathcal{P} = \mathcal{P}_2 \otimes \mathcal{D}(s_3) \text{ for some } \mathcal{P}_2 \text{ such that } s_1\ \mathcal{R}\ \mathcal{P}_2. \tag{8.4}$$

Condition 1 of the definition of a probabilistic forward simulation is immediate to verify. Condition 2 for transitions that involve M_1 only or M_3 only is immediate to verify as well.

Let $(s_1, s_3)\ \mathcal{R}'\ \mathcal{P}_2 \otimes \mathcal{D}(s_3)$, and let $(s_1, s_3) \xrightarrow{a} \mathcal{P}_1 \otimes \mathcal{P}_3$, where $s_1 \xrightarrow{a} \mathcal{P}_1$, and $s_3 \xrightarrow{a} \mathcal{P}_3$. From the definition of a probabilistic forward simulation, for each $s \in \Omega_2$ there exists a weak combined transition $s_2 \overset{a}{\Longrightarrow}_C \mathcal{P}_s$ of M_2, and there exists a probability space $\mathcal{P}'_2$ of $Probs(Probs(states(M_2)))$, such that

$$\sum_{s \in \Omega_2} P_2[s]\mathcal{P}_s = \sum_{\mathcal{P} \in \Omega'_2} P'_2[\mathcal{P}]\mathcal{P}, \tag{8.5}$$

and

$$\mathcal{P}_1 \sqsubseteq_{\mathcal{R}} \mathcal{P}'_2. \tag{8.6}$$

For each $s \in \Omega_2$, let $\mathcal{O}_s$ be a generator for $s \overset{a}{\Longrightarrow}_C \mathcal{P}_s$. Define a new generator $\mathcal{O}'_s$ as follows: for each finite execution fragment α of $M_2 \| M_3$ starting in (s, s_3),

1. if $\mathcal{O}_s(\alpha\lceil M_2) = (s', \mathcal{P})$, where $(s', \mathcal{P}) = \sum_i p_i(s', a_i, \mathcal{P}_i)$, each $(s', a_i, \mathcal{P}_i)$ is a transition of M_2, and $\alpha\lceil M_3 = s_3$, then

   $$\mathcal{O}'_s(\alpha) = \sum_i p_i((s', s_3), a_i, \mathcal{P}_i \otimes \mathcal{P}'_i),$$

   where

   $$\mathcal{P}'_i = \mathcal{D}(s_3) \text{ if } a_i \neq a, \text{ and } \mathcal{P}'_i = \mathcal{P}_3 \text{ if } a_i = a.$$

2. if $\mathcal{O}_s(\alpha\lceil M_2) = (s', \mathcal{P})$, where $(s', \mathcal{P}) = \sum_i p_i(s', a_i, \mathcal{P}_i)$, each $(s', a_i, \mathcal{P}_i)$ is a transition of M_2, $\alpha\lceil M_3 = s_3 a s'_3$, and $s'_3 \in \Omega_3$, then

   $$\mathcal{O}'_s(\alpha) = \sum_i p_i((s', s'_3), a_i, \mathcal{P}_i \otimes \mathcal{D}(s'_3));$$

3. if none of the above cases holds, then $\mathcal{O}'_s(\alpha) = \mathcal{D}(\delta)$.

The weak combined transition generated by each $\mathcal{O}'_s$ is $(s, s_3) \overset{a}{\Longrightarrow}_{\mathrm{C}} \mathcal{P}_s \otimes \mathcal{P}_3$. In fact, an execution fragment α of $M_2 \| M_3$ is terminal for $\mathcal{O}'_s$ iff $\alpha \lceil M_2$ is terminal for $\mathcal{O}_s$ and $\alpha \lceil M_3 = s_3 a s'_3$ for $s'_3 \in \Omega_3$, and thus $\Omega_{\mathcal{O}'_s} = \Omega_s \times \Omega_3$. Moreover, for each $\alpha \in \Omega_{\mathcal{O}'_s}$, $P_\alpha^{\mathcal{O}'_s} = P_{\alpha \lceil M_2}^{\mathcal{O}_s} P_3[lstate(\alpha \lceil M_3)]$.

Denote $\mathcal{P}_s \otimes \mathcal{P}_3$ by $\mathcal{P}_{(s,s_3)}$. Then, for each $(s, s_3) \in \Omega_2 \times \mathcal{D}(s_3)$, we have identified a weak combined transition $(s, s_3) \overset{a}{\Longrightarrow}_{\mathrm{C}} \mathcal{P}_{(s,s_3)}$. These are the spaces of Condition 2.a in the definition of a probabilistic forward simulation. Note that $\mathcal{P}_{(s,s_3)}$ can be expressed alternatively as

$$\mathcal{P}_{(s,s_3)} = \sum_{s'_3 \in \Omega_3} P_3[s'_3] \left(\mathcal{P}_s \otimes \mathcal{D}(s'_3) \right). \tag{8.7}$$

Let

$$\mathcal{P}'_{2,3} \triangleq \sum_{s'_3 \in \Omega_3} P_3[s'_3] \left(\mathcal{P}'_2 \otimes \mathcal{D}(\mathcal{D}(s'_3)) \right), \tag{8.8}$$

where the pairing of two probability spaces is meant to be their product. For each $s'_3 \in \Omega_3$, since $\mathcal{P}_1 \sqsubseteq_{\mathcal{R}} \mathcal{P}'_2$, $\mathcal{P}_1 \otimes \mathcal{D}(s'_3) \sqsubseteq_{\mathcal{R}} \mathcal{P}'_2 \otimes \mathcal{D}(\mathcal{D}(s'_3))$. Thus, from Lemma 8.2.1, $\mathcal{P}_1 \otimes \mathcal{P}_3 \sqsubseteq_{\mathcal{R}} \mathcal{P}'_{2,3}$. This is enough to show that Condition 2.b of the definition of a probabilistic forward simulation is satisfied.

We are left with $\sum_{s \in \Omega_2} P_2[s] \mathcal{P}_{(s,s_3)} = \sum_{\mathcal{P} \in \Omega'_{2,3}} P'_{2,3}[\mathcal{P}] \mathcal{P}$, which is shown as follows. From (8.7),

$$\sum_{s \in \Omega_2} P_2[s] \mathcal{P}_{(s,s_3)} = \sum_{s \in \Omega_2} \sum_{s'_3 \in \Omega_3} P_2[s] P_3[s'_3] \left(\mathcal{P}_s \otimes \mathcal{D}(s'_3) \right). \tag{8.9}$$

From (8.5),

$$\sum_{s \in \Omega_2} P_2[s] \mathcal{P}_{(s,s_3)} = \sum_{s'_3 \in \Omega_3} \sum_{\mathcal{P} \in \Omega'_2} P'_2[\mathcal{P}] P_3[s'_3] \left(\mathcal{P} \otimes \mathcal{D}(s'_3) \right). \tag{8.10}$$

From a simple algebraic manipulation,

$$\sum_{s \in \Omega_2} P_2[s] \mathcal{P}_{(s,s_3)} = \sum_{s'_3 \in \Omega_3} \sum_{\mathcal{P} \in \Omega_{\mathcal{P}'_2 \otimes \mathcal{D}(\mathcal{D}(s'_3))}} P_3[s'_3] P'_2[\mathcal{P}] \mathcal{P}. \tag{8.11}$$

From (8.8),

$$\sum_{s \in \Omega_2} P_2[s] \mathcal{P}_{(s,s_3)} = \sum_{\mathcal{P} \in \Omega'_{2,3}} P'_{2,3}[\mathcal{P}] \mathcal{P}. \tag{8.12}$$

■

8.6 The Execution Correspondence Theorem

The existence of some simulation relation between two probabilistic automata implies that there is some strict relation between their probabilistic executions. This relationship is known as the *execution correspondence lemma* for ordinary automata [GSSL94] and is useful in the context of liveness. In this section we prove the execution correspondence theorem for probabilistic automata; a corollary, which is proved in Section 8.7, is that the existence of a probabilistic forward simulation is sound for the trace distribution precongruence.

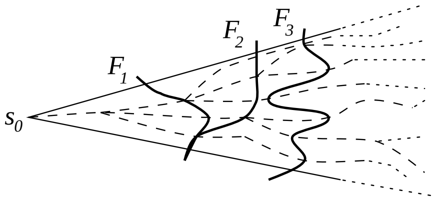


Figure 8-6: Fringes.

8.6.1 Fringes

Let H be a probabilistic execution of a probabilistic automaton M. Define the extended states of H, denoted by *extstates*(H), to be *states*$(H) \cup \{q\delta \mid q \in$ *states*$(H), P_H[C_{q\delta}] > 0\}$. A *fringe* of H is a discrete probability space $\mathcal{P}$ of *Probs*(*extstates*(H)) such that for each state q of H,

$$\sum_{q' \in \Omega \mid q \leq q'} P[q'] \leq P_H[C_q]. \tag{8.13}$$

Two fringes $\mathcal{P}_1$ and $\mathcal{P}_2$ are in the $\leq$ relation iff for each state q of H,

$$\sum_{q' \in \Omega_1 \mid q \leq q'} P_1[q'] \leq \sum_{q' \in \Omega_2 \mid q \leq q'} P_2[q']. \tag{8.14}$$

Informally, a fringe is a line that cuts a probabilistic execution in two parts (see Figure 8-6). A fringe is smaller than another one if the first fringe cuts the probabilistic execution earlier than the second fringe. Figure 8-6 shows three fringes F_1, F_2 and F_3, where $F_1 \leq F_2 \leq F_3$.

A fringe of particular interest is the fringe that cuts a probabilistic execution fragment at some depth i. Let *fringe*(H, i) denote the fringe of H where $\Omega = \{q \in$ *extstates*$(H) \mid |q| = i\} \cup \{q\delta \in$ *extstates*$(H) \mid |q| < i\}$, and for each $q \in \Omega$, $P[q] = P_H[C_q]$.

8.6.2 Execution Correspondence Structure

Let $\mathcal{R}$ be a probabilistic forward simulation from M_1 to M_2. An *execution correspondence structure* via $\mathcal{R}$ is a tuple (H_1, H_2, m, S), where H_1 is a probabilistic execution of M_1, H_2 is a probabilistic execution of M_2, m is a mapping from natural numbers to fringes of M_2, and S is a mapping from natural numbers to probability distributions of *Probs*(*Probs*(*states*(H_2))), such that

1. For each i, $m(i) \leq m(i+1)$;
2. For each state q_2 of H_2, $\lim_{i\to\infty} \sum_{q \in \Omega_i \mid q_2 \leq q} P_i[q] = P_H[C_q]$;
3. Let $q_1 \; \mathcal{R} \; \mathcal{P}$ iff for each $q \in \Omega$, *trace*$(q) =$ *trace*(q_1), and either
   (a) q_1 does not end in δ, each state of Ω does not end in δ, and *lstate*$(q_1) \; \mathcal{R}$ *lstate*$(\mathcal{P})$, or
   (b) q_1 and each state of Ω end in δ and *lstate*(δ-*strip*$(q_1)) \; \mathcal{R}$ *lstate*(δ-*strip*$(\mathcal{P})$).

   Then, for each $i \geq 0$, $m(i) = \sum_{\mathcal{P} \in \Omega_{S(i)}} P_{S(i)}[\mathcal{P}]\mathcal{P}$, and *fringe*$(H_1, i) \sqsubseteq_{\mathcal{R}} S(i)$.

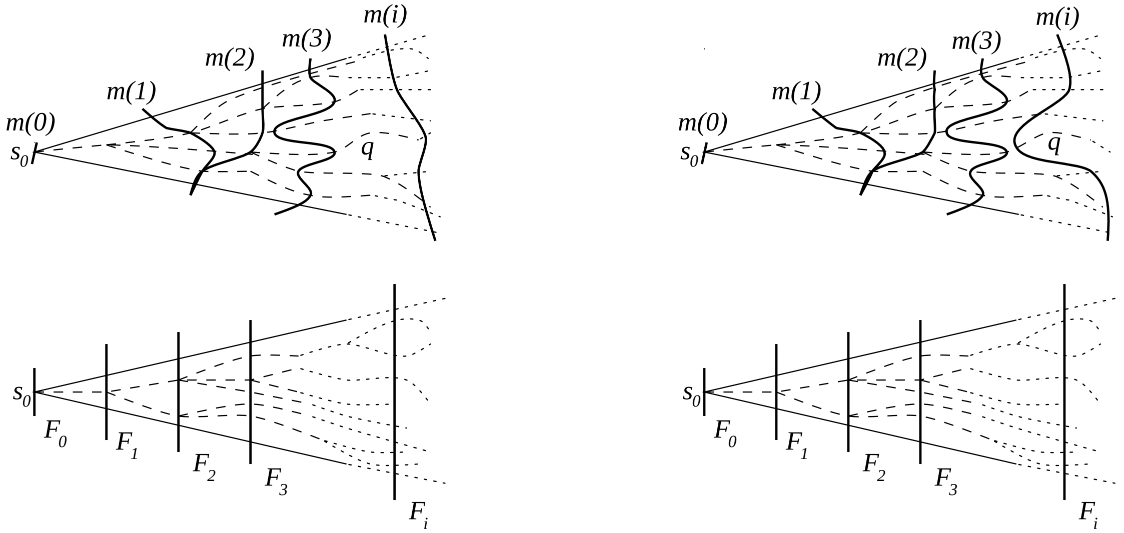


Figure 8-7: Execution Correspondence Structures: the role of Condition 2.

4. Let, for each $i \geq 0$, each $q_1 \in \mathit{fringe}(H_1, i)$, and each $q_2 \in \mathit{states}(H_2)$, $W_i(q_1, q_2) \triangleq \sum_{\mathcal{P}} w_i(q_1, \mathcal{P})P[q_2]$. If $W_i(q_1, q_2') = 0$ for each prefix or extension q_2' of q_2, then, for each extension q_1' of q_1 such that $q_1' \in \mathit{fringe}(H_1, i+1)$ and each prefix or extension q_2' of q_2, $W_{i+1}(q_1', q_2') = 0$.

Informally, an execution correspondence structure is an object that shows how a probabilistic execution H_1 of M_1 is represented by a probabilistic execution H_2 of M_2 via $\mathcal{R}$. H_2 is said to be the probabilistic execution fragment that corresponds to H_1. Conditions 1 and 3 state that each fringe $\mathit{fringe}(H_1, i)$ is represented by the fringe $m(i)$ in H_2, and Condition 2 states that at the limit each state of H_2 represents some part of H_1. Figure 8-7 gives an example of an execution correspondence structure (left) and of a structure that fails to satisfy Condition 2 since state q is not captured (right). Condition 4 enforces the correspondence between H_1 and H_2. Informally, it states that if two states q_1 and q_2 of H_1 and H_2, respectively, are connected through the i^{th} fringes, then for each $j < i$ there are two prefixes q_1' and q_2' of q_1 and q_2, respectively, that are connected through the j^{th} fringes. This condition allows us to derive a correspondence structure between the execution fragments of M_1 and M_2 that denote the states of H_1 and H_2. We do not use Condition 4 to prove any of the results that we present in this thesis; however, this condition is necessary to prove the results that Segala and Lynch present in [SL94].

If $\mathcal{R}$ is a weak probabilistic simulation, then an execution correspondence structure is a triplet (H_1, H_2, m): Condition 3 becomes $\mathit{fringe}(H_1, i) \sqsubseteq_{\mathcal{R}} m(i)$, where $q_1 \mathrel{\mathcal{R}} q_2$ iff $\mathit{trace}(q_1) = \mathit{trace}(q_2)$ and either q_1 and q_2 end in δ and $\delta\text{-}\mathit{strip}(\mathit{lstate}(q_1)) \mathrel{\mathcal{R}} \delta\text{-}\mathit{strip}(\mathit{lstate}(q_2))$, or $\mathit{lstate}(q_1) \mathrel{\mathcal{R}} \mathit{lstate}(q_2)$; $W_i(q_1, q_2)$ becomes $w_i(q_1, q_2)$, and Condition 4 says that for each $i \geq 0$, given $q_1 \in \mathit{fringe}(H_1, i)$ and $q_2 \in \mathit{states}(H_2)$, if $w_i(q_1, q_2') = 0$ for each prefix or extension q_2' of q_2, then, for each extension q_1' of q_1 such that $q_1' \in \mathit{fringe}(H_1, i+1)$, and each prefix or extension q_2' of q_2, $w_{i+1}(q_1', q_2') = 0$.

If $\mathcal{R}$ is a strong probabilistic simulation, then an execution correspondence structure is a pair (H_1, H_2): Conditions 1 and 2 are removed; Condition 3 becomes $\mathit{fringe}(H_1, i) \sqsubseteq_{\mathcal{R}} \mathit{fringe}(H_2, i)$ where $q_1 \mathrel{\mathcal{R}} q_2$ iff $\mathit{itrace}(q_1) = \mathit{itrace}(q_2)$ and either q_1 and q_2 end in δ and $\delta\text{-}\mathit{strip}(\mathit{lstate}(q_1)) \mathrel{\mathcal{R}} \delta\text{-}\mathit{strip}(\mathit{lstate}(q_2))$, or $\mathit{lstate}(q_1) \mathrel{\mathcal{R}} \mathit{lstate}(q_2)$; Condition 4 says that for each $i \geq 0$, given $q_1 \in \mathit{fringe}(H_1, i)$ and $q_2 \in \mathit{fringe}(H_2, i)$, if $w_i(q_1, q_2) = 0$, then, for each extension q_1' of q_1 such that $q_1' \in \mathit{fringe}(H_1, i{+}1)$ and each extension q_2' of q_2 such that $q_2' \in \mathit{fringe}(H_2, i{+}1)$, $w_{i+1}(q_1', q_2') = 0$.

8.6.3 The Main Theorem

Theorem 8.6.1 *Let $M_1 \sqsubseteq_{FS} M_2$ via the probabilistic forward simulation $\mathcal{R}$, and let H_1 be a probabilistic execution of M_1. Then there exists a probabilistic execution H_2 of M_2, a mapping m from natural numbers to fringes of M_2, and a mapping S from natural numbers to probability distributions of $\mathit{Probs}(\mathit{Probs}(\mathit{states}(H_2)))$, such that (H_1, H_2, m, S) is an execution correspondence structure via $\mathcal{R}$.*

Proof. Let q_1 be a state of H_1, and let $\mathcal{P}_2$ be a distribution over potential states of H_2 such that $q_1 \sqsubseteq_{\mathcal{R}} \mathcal{P}_2$ according to the definition given in the definition of an execution correspondence structure. Denote by $\mathcal{P}_{H_1}^{q_1}$ the probability space such that $tr_{q_1}^{H_1} = \sum_{tr \in \Omega_{H_1}^{q_1}} P_{H_1}^{q_1}[tr](q_1 \frown tr)$. Let $tr_1 \in \Omega_{H_1}^{q_1}$, and let $\mathcal{P}_{tr_1}$ be the probability space reached in $q_1 \frown tr_1$.

Since $\mathcal{R}$ is a probabilistic forward simulation, then for each state q_2 of Ω_2 there exists a weak transition $tr_{q_1\mathcal{P}_2 tr_1 q_2}$ of H_2 with action $a \restriction \mathit{ext}(M_2)$, leading to a distribution over states $\mathcal{P}_{q_1\mathcal{P}_2 tr_1 q_2}$, such that there exists a probability distribution over probability distributions of potential states of H_2, denoted by $\mathcal{P}_{q_1\mathcal{P}_2 tr_1}^S$, satisfying

$$\sum_{\mathcal{P} \in \Omega_{q_1\mathcal{P}_2 tr_1}^S} P_{q_1\mathcal{P}_2 tr_1}^S[\mathcal{P}]\mathcal{P} = \sum_{q_2 \in \Omega_2} P_2[q_2]\mathcal{P}_{q_1\mathcal{P}_2 tr_1 q_2} \tag{8.15}$$

and

$$\mathcal{P}_{tr_1} \sqsubseteq_{\mathcal{R}} \mathcal{P}_{q_1\mathcal{P}_2 tr_1}^S \tag{8.16}$$

via a weight function $w_{q_1\mathcal{P}_2 tr_1}$. Denote the probability space $\sum_{q_2 \in \Omega_2} P_2[q_2]\mathcal{P}_{q_1\mathcal{P}_2 tr_1 q_2}$ by $\mathcal{P}_{q_1\mathcal{P}_2 tr_1}$, i.e.,

$$\mathcal{P}_{q_1\mathcal{P}_2 tr_1} \triangleq \sum_{q_2 \in \Omega_2} P_2[q_2]\mathcal{P}_{q_1\mathcal{P}_2 tr_1 q_2}. \tag{8.17}$$

Denote the generator of each weak transition $tr_{q_1\mathcal{P}_2 tr_1 q_2}$ by $\mathcal{O}_{q_1\mathcal{P}_2 tr_1 q_2}$ (cf. Section 4.2.7). For the sake of this proof, we change the notation for the generators of the transitions of a probabilistic execution. Thus, for each q_2' such that $q_2 \leq q_2'$, $\mathcal{O}_{q_1\mathcal{P}_2 tr_1 q_2}(q_2')$ stands for $\mathcal{O}_{q_1\mathcal{P}_2 tr_1 q_2}(q_2' {\uparrow} q_2)$, and $P_{q_2'}^{\mathcal{O}_{q_1\mathcal{P}_2 tr_1 q_2}}$ stands for $P_{q_2' \uparrow q_2}^{\mathcal{O}_{q_1\mathcal{P}_2 tr_1 q_2}}$.

For each state q_1 and each probability distribution over states $\mathcal{P}_2$, let $\delta_{q_1} \triangleq \mathcal{D}(q_1\delta)$, $\delta_{\mathcal{P}_2} \triangleq \sum_{q_2 \in \Omega_2} P_2[q_2]\delta_{q_2}$, $\delta_{\mathcal{P}_2}^S \triangleq \mathcal{D}(\delta_{\mathcal{P}_2})$, and $w_{\delta q_1 \mathcal{P}_2}$ be a weight function such that $w_{\delta q_1 \mathcal{P}_2}(q_1\delta, \mathcal{P}_2) = 1$. Note that, if for each $q_2 \in \Omega_2$, $\mathit{trace}(q_1) = \mathit{trace}(q_2)$, then

$$\delta_{q_1} \sqsubseteq_{\mathcal{R}} \delta_{\mathcal{P}_2}^S \tag{8.18}$$

via $w_{\delta q_1 \mathcal{P}_2}$. Moreover,

$$\delta_{\mathcal{P}_2} = \sum_{\mathcal{P} \in \Omega^S_{\mathcal{P}_2}} P_{\delta^S_{\mathcal{P}_2}}[\mathcal{P}]\mathcal{P}. \tag{8.19}$$

Let s_1 be the start state of H_1, and s_2 be a start state of M_2 that is related to s_1. We know that s_2 exists since $\mathcal{R}$ is a probabilistic forward simulation. Let *Active* be the smallest set such that

1. $(s_1, \mathcal{D}(s_2)) \in \mathit{Active}$;
2. if $(q_1, \mathcal{P}_2) \in \mathit{Active}$, $tr_1 \in \Omega^{q_1}_{H_1}$, and $(q_1', \mathcal{P}_2') \in \Omega_{tr_1} \times \Omega^S_{q_1 \mathcal{P}_2 tr_1}$, then $(q_1', \mathcal{P}_2') \in \mathit{Active}$;
3. if $(q_1, \mathcal{P}_2) \in \mathit{Active}$, $P^{q_1}_{H_1}[\delta] > 0$, then $(q_1\delta, \delta^S_{\mathcal{P}_2}) \in \mathit{Active}$.

Observe that for each pair $(q_1, \mathcal{P}_2) \in \mathit{Active}$, $q_1 \;\mathcal{R}\; \mathcal{P}_2$ (simple inductive argument). For each q_1 such that there exists some $\mathcal{P}_2$ with $(q_1, \mathcal{P}_2) \in \mathit{Active}$, each $tr_1 \in \Omega^{q_1}_{H_1}$, and each $q_2 \in \Omega_2$, let $\mathit{active}(q_1, \mathcal{P}_2, tr_1, q_2)$ be the set of states that are active in $\mathcal{O}_{q_1 \mathcal{P}_2 tr_1 q_2}$, and let $\mathit{reach}(q_1, \mathcal{P}_2, tr_1, q_2)$ be the set of states that are reachable in $\mathcal{O}_{q_1 \mathcal{P}_2 tr_1 q_2}$. Let *active* denote the union of the sets $\mathit{reach}(q_1, \mathcal{P}_2, tr_1, q_2)$ where $(q_1, \mathcal{P}_2) \in \mathit{Active}$, $tr_1 \in \Omega^{q_1}_{H_1}$, and $q_2 \in \Omega_2$. For each $i \leq 0$, let $\mathit{Active}(i)$ be the set of pairs $(q_1, \mathcal{P}_2) \in \mathit{Active}$ such that either $|q_1| = i$ or $|q_1| \leq i$ and q_1 ends in δ. For each pair $(q_1, \mathcal{P}_2)$ of *Active* such that q_1 does not end in δ, let

$$\mathcal{P}_{q_1} \;\triangleq\; \sum_{tr_1 \in \Omega^{q_1}_{H_1}} P^{q_1}_{H_1}[tr_1]\mathcal{P}_{tr_1} + P^{q_1}_{H_1}[\delta]\delta_{q_1} \tag{8.20}$$

be the probability space reached in H_1 with the transition enabled from q_1,

$$\mathcal{P}_{q_1 \mathcal{P}_2} \;\triangleq\; \sum_{tr_1 \in \Omega^{q_1}_{H_1}} P^{q_1}_{H_1}[tr_1]\mathcal{P}_{q_1 \mathcal{P}_2 tr_1} + P^{q_1}_{H_1}[\delta]\delta_{\mathcal{P}_2} \tag{8.21}$$

be the probability space that is reached in the corresponding transition of $\mathcal{P}_2$,

$$\mathcal{P}^S_{q_1 \mathcal{P}_2} \;\triangleq\; \sum_{tr_1 \in \Omega^{q_1}_{H_1}} P^{q_1}_{H_1}[tr_1]\mathcal{P}^S_{q_1 \mathcal{P}_2 tr_1} + P^{q_1}_{H_1}[\delta]\delta^S_{\mathcal{P}_2} \tag{8.22}$$

be the probability space of probability spaces that corresponds to $\mathcal{P}_{q_1}$, and for each $q_1', \mathcal{P}_2'$,

$$w_{q_1 \mathcal{P}_2}(q_1', \mathcal{P}_2') \;\triangleq\; \sum_{tr_1 \in \Omega^{q_1}_{H_1}} P^{q_1}_{H_1}[tr_1] w_{q_1 \mathcal{P}_2 tr_1}(q_1', \mathcal{P}_2') + P^{q_1}_{H_1}[\delta] w_{\delta q_1 \mathcal{P}_2}(q_1', \mathcal{P}_2') \tag{8.23}$$

be the corresponding weight function. From Lemma 8.2.1,

$$\mathcal{P}_{q_1} \sqsubseteq_{\mathcal{R}} \mathcal{P}^S_{q_1 \mathcal{P}_2} \tag{8.24}$$

via the weight function $w_{q_1 \mathcal{P}_2}$.
For each pair $(q_1, \mathcal{P}_2)$ of *Active* such that q_1 ends in δ, let

$$\mathcal{P}_{q_1} \;\triangleq\; \mathcal{D}(q_1), \quad \mathcal{P}_{q_1, \mathcal{P}_2} \;\triangleq\; \mathcal{P}_2, \quad \mathcal{P}^S_{q_1, \mathcal{P}_2} \;\triangleq\; \mathcal{D}(\mathcal{P}_2), \quad \text{and } w_{q_1 \mathcal{P}_2}(q_1, \mathcal{P}_2) \;\triangleq\; 1. \tag{8.25}$$

It is immediate to observe that Equation (8.24) holds also in this case.

Define $m(i), S(i)$ and w_i inductively as follows.

$$m(0) \triangleq \mathcal{D}(s_2), \quad S(0) \triangleq \mathcal{D}(m(0)), \quad w_0(s_1, m(0)) \triangleq 1, \tag{8.26}$$

$$m(i+1) \triangleq \sum_{(q_1,\mathcal{P}_2)\in Active(i)} w_i(q_1,\mathcal{P}_2)\mathcal{P}_{q_1\mathcal{P}_2}, \tag{8.27}$$

$$S(i+1) \triangleq \sum_{(q_1,\mathcal{P}_2)\in Active(i)} w_i(q_1,\mathcal{P}_2)\mathcal{P}^S_{q_1\mathcal{P}_2}, \tag{8.28}$$

$$w_{i+1}(q_1',\mathcal{P}_2') \triangleq \sum_{(q_1,\mathcal{P}_2)\in Active(i)} w_i(q_1,\mathcal{P}_2)w_{q_1\mathcal{P}_2}(q_1',\mathcal{P}_2'). \tag{8.29}$$

To show that Equations (8.27), (8.28),and (8.29) are well defined, we show by induction that for each $i \geq 0$, $\sum_{(q_1,\mathcal{P}_2)\in Active(i)} w_i(q_1,\mathcal{P}_2) = 1$. The base case is a direct consequence of (8.26) and the definition of *Active*(0). For the inductive step,

$$\begin{aligned}
&\sum_{(q_1,\mathcal{P}_2)\in Active(i+1)} w_{i+1}(q_1,\mathcal{P}_2)\\
&= \sum_{(q_1,\mathcal{P}_2)\in Active(i+1)} \sum_{(q_1',\mathcal{P}_2')\in Active(i)} w_i(q_1',\mathcal{P}_2')w_{q_1'\mathcal{P}_2'}(q_1,\mathcal{P}_2)\\
&= \sum_{(q_1',\mathcal{P}_2')\in Active(i)} w_i(q_1',\mathcal{P}_2')\\
&= 1,
\end{aligned}$$

where the first step follows from Equation (8.29), the second step follows from the fact that $w_{q_1',\mathcal{P}_2'}$ is a weight function that is non zero only in pairs of *Active*$(i+1)$, and the third step follows from induction. Let

$$W_{q_1\mathcal{P}_2 tr_1 q_2}(q_2') \triangleq w(q_1,\mathcal{P}_2)P_{H_1}^{q_1}[tr_1]P_2[q_2]P_{q_2'}^{\mathcal{O}_{q_1\mathcal{P}_2 tr_1 q_2}}. \tag{8.30}$$

Consider a state q_2 of *active*. Then the transition enabled from q_2 is

$$\sum_{(q_1',\mathcal{P}_2')\in Active} \sum_{tr_1\in\Omega^{H_1}_{q_1'}} \sum_{q_2'\in\Omega_2'|q_2\in active(q_1',\mathcal{P}_2',tr_1,q_2')} \tag{8.31}$$

$$P_{\mathcal{O}_{q_1\mathcal{P}_2 tr_1 q_2'}(q_2)}[acts(M_2)]W_{q_1'\mathcal{P}_2' tr_1 q_2'}(q_2)/W(q_2)\left(\mathcal{O}_{q_1'\mathcal{P}_2 tr_1 q_2'}(q_2) \restriction acts(M_2)\right),$$

where $W(s_2) \triangleq 1$, and for each $q_2 \neq s_2$,

$$W(q_2) \triangleq \sum_{(q_1',\mathcal{P}_2')\in Active} \sum_{tr_1\in\Omega^{H_1}_{q_1'}} \sum_{q_2'\in\Omega_2'|q_2'\neq q_2, q_2\in reach(q_1',\mathcal{P}_2',tr_1,q_2')} W_{q_1'\mathcal{P}_2' tr_1 q_2'}(q_2). \tag{8.32}$$

It is easy to verify that Expression (8.31) denotes a valid transition of a probabilistic execution fragment of M since it is the combination of legal transitions of a probabilistic execution fragment of M. The fact that the projection of a legal transition of a probabilistic execution fragment of M onto $acts(M)$ is still a legal transition of a probabilistic execution fragment of M follows from the fact that M is a simple probabilistic automaton.

Informally, the set *active* is used to identify all the states of H_2. The transition enabled from each one of those states, say q_2, is due to several states of H_1, and each state of H_1 influences the transition enabled from a specific state of H_2 with a different probability. Such a probability depends on how much a state of H_2 represents a state of H_1, on the probability of the transition of M_1 that has to be matched, on the probability of reaching a specific state q_2' of H_2 during the matching operation, on the probability of reaching q_2 from q_2', and on the probability of departing from q_2. These conditions are captured by $P_{\mathcal{O}_{q_1\mathcal{P}_2 tr_1 q_2'}(q_2)}[acts(M_2)]W_{q_1'\mathcal{P}_2' tr_1 q_2'}(q_2)$. These weights must be normalized with respect to the probability of reaching q_2, which is expressed by $W(q_2)$. The condition $q_2' \neq q_2$ in the third sum of (8.32) is justified by the fact $W(q_2)$ is the probability of reaching q_2.

This completes the definition of H_2, $m(i)$, $S(i)$, and the w_i's. We need to show that (H_1, H_2, w, S) is an execution correspondence structure via $\mathcal{R}$. Thus, we need to show the following properties.

1. For each i, $m(i)$ is a fringe of H_2;
2. For each i, $m(i) \leq m(i+1)$;
3. For each state q of H_2, $\lim_{i\to\infty} \sum_{q'\in\Omega_i|q\leq q'} P_i[q'] = P_H[C_q]$;
4. For each i, $m(i) = \sum_{\mathcal{P}\in S(i)} P_{S(i)}[\mathcal{P}]\mathcal{P}$;
5. For each i, $\mathit{fringe}(H_1, i) \sqsubseteq_{\mathcal{R}} S(i)$ via w_i.
6. For each i, each $q_1 \in \mathit{fringe}(H_1, i)$, and each $q_2 \in \mathit{states}(H_2)$, if $W_i(q_1, q_2') = 0$ for each prefix or extension q_2' of q_2, then, for each extension q_1' of q_1 such that $q_1' \in \mathit{fringe}(H_1, i+1)$ and each prefix or extension q_2' of q_2, $W_{i+1}(q_1', q_2') = 0$.

We show each item separately.

1. For each i, $m(i)$ is a fringe of H_2.

   By construction $m(i)$ is a probability distribution. Thus, we need to show only that for each state q_2 of H_2,

   $$\sum_{q_2'\in\Omega_{m(i)}|q_2\leq q_2'} P_{m(i)}[q_2'] \leq P_{H_2}[C_{q_2}] \tag{8.33}$$

   First we show that for each $q_2 \in \mathit{states}(H_2)$,

   $$W(q_2) = P_{H_2}[C_{q_2}]; \tag{8.34}$$

   then we show that for each $q_2 \in \mathit{states}(H_2)$,

   $$\sum_{q_2'\in\Omega_{m(i)}|q_2\leq q_2'} P_{m(i)}[q_2'] \leq W(q_2). \tag{8.35}$$

   The proof of (8.34) is by induction on the length of q_2. If $q_2 = s_2$, then (8.34) holds by definition. Otherwise, let $\tilde{q}_2$ be q_2 without its last action and state, i.e., $q_2 = \tilde{q}_2 as$ for some action a and some state s. Then, from the definition of the probability of a cone, induction, Equation (8.31) and an algebraic simplification,

$$P_{H_2}[C_{q_2}] = \sum_{(q_1',\mathcal{P}_2')\in Active} \sum_{tr_1\in\Omega_{H_1}^{q_1'}} \sum_{q_2'\in\Omega_2' \mid \tilde{q}_2\in active(q_1',\mathcal{P}_2',tr_1,q_2')}$$

$$W_{q_1'\mathcal{P}_2' tr_1 q_2'}(\tilde{q}_2) P_{\mathcal{O}_{q_1'\mathcal{P}_2 tr_1 q_2'}(\tilde{q}_2)}[q_2]. \tag{8.36}$$

From Equation (8.30) and the definition of $P_{q_2}^{\mathcal{O}_{q_1'\mathcal{P}_2' tr_1 q_2'}}$ (cf. Section 4.2.7), we obtain

$$P_{H_2}[C_{q_2}] = \sum_{(q_1',\mathcal{P}_2')\in Active} \sum_{tr_1\in\Omega_{H_1}^{q_1'}} \sum_{q_2'\in\Omega_2' \mid \tilde{q}_2\in active(q_1',\mathcal{P}_2',tr_1,q_2')}$$

$$w(q_1', \mathcal{P}_2') P_{H_1}^{q_1'}[tr_1] P_2'[q_2'] P_{q_2}^{\mathcal{O}_{q_1'\mathcal{P}_2' tr_1 q_2'}}. \tag{8.37}$$

Observe that $q_2' \in \Omega_2'$ and $\tilde{q}_2 \in active(q_1', \mathcal{P}_2', tr_1, q_2')$ iff $q_2' \in \Omega_2'$, $q_2' \neq q_2$, and $q_2 \in reach(q_1', \mathcal{P}_2', tr_1, q_2')$. Thus, from Equation (8.31),

$$P_{H_2}[C_{q_2}] = \sum_{(q_1',\mathcal{P}_2')\in Active} \sum_{tr_1\in\Omega_{H_1}^{q_1'}} \sum_{q_2'\in\Omega_2' \mid q_2'\neq q_2, q_2\in reach(q_1',\mathcal{P}_2',tr_1,q_2')} W_{q_1'\mathcal{P}_2' tr_1 q_2'}(q_2). \tag{8.38}$$

At this point Equation (8.32) is sufficient to conclude the validity of Equation (8.34).

The proof of Equation (8.35) is also by induction. If $i = 0$, then the result follows directly from the fact that a fringe is a probability distribution. Otherwise, let $N(q_1)$ be true iff q_1 does not end in δ. Then, from Equation (8.27),

$$\sum_{q_2'\in\Omega_{m(i+1)} \mid q_2\leq q_2'} P_{m(i+1)}[q_2'] \tag{8.39}$$

can be rewritten into

$$\sum_{q_2'\in\Omega_{m(i+1)} \mid q_2\leq q_2'} \sum_{(q_1,\mathcal{P}_2)\in Active(i)} w_i(q_1, \mathcal{P}_2) P_{q_1\mathcal{P}_2}[q_2']. \tag{8.40}$$

From the definition of $\mathcal{P}_{q_1,\mathcal{P}_2}$ (Equations (8.21) and (8.25)) and the definition of $\mathcal{P}_{q_1\mathcal{P}_2 tr_1}$ (Equation (8.17)), Expression (8.40) can be rewritten into

$$\sum_{q_2'\in\Omega_{m(i+1)} \mid q_2\leq q_2'} \sum_{(q_1,\mathcal{P}_2)\in Active(i), N(q_1)} \sum_{tr_1\in\Omega_{H_1}^{q_1}} \sum_{q_2''\in\Omega_2} \tag{8.41}$$

$$w_i(q_1, \mathcal{P}_2) P_{H_1}^{q_1}[tr_1] P_2[q_2''] P_{q_1\mathcal{P}_2 tr_1 q_2''}[q_2']$$

$$+ \sum_{q_2'\delta\in\Omega_{m(i+1)} \mid q_2\leq q_2'} \sum_{(q_1,\mathcal{P}_2)\in Active(i), N(q_1)} w_i(q_1, \mathcal{P}_2) P_{H_1}^{q_1}[\delta] P_2[q_2']$$

$$+ \sum_{q_2'\delta\in\Omega_{m(i+1)} \mid q_2\leq q_2'} \sum_{(q_1\delta,\mathcal{P}_2)\in Active(i)} w_i(q_1\delta, \mathcal{P}_2) P_2[q_2'\delta].$$

By exchanging sums in Expression (8.41), we obtain

$$\begin{aligned}&\sum_{(q_1,\mathcal{P}_2)\in Active(i),N(q_1)}\ \sum_{tr_1\in\Omega^{q_1}_{H_1}}\ \sum_{q_2''\in\Omega_2}\ \sum_{q_2'\in\Omega_{m(i+1)}|q_2\leq q_2'}\\&\qquad w_i(q_1,\mathcal{P}_2)P^{q_1}_{H_1}[tr_1]P_2[q_2'']P_{q_1\mathcal{P}_2tr_1q_2''}[q_2']\\&+\sum_{(q_1,\mathcal{P}_2)\in Active(i),N(q_1)}\ \sum_{q_2'\delta\in\Omega_{m(i+1)}|q_2\leq q_2'} w_i(q_1,\mathcal{P}_2)P^{q_1}_{H_1}[\delta]P_2[q_2']\\&+\sum_{(q_1\delta,\mathcal{P}_2)\in Active(i)}\ \sum_{q_2'\delta\in\Omega_{m(i+1)}|q_2\leq q_2'} w_i(q_1\delta,\mathcal{P}_2)P_2[q_2'\delta],\end{aligned}\tag{8.42}$$

where the first summand comes from the first summand of (8.22), the second summand comes from the second summand of (8.22), and the third summand comes from (8.25). Consider the first summand of Expression (8.42), and partition the states q_2'' of Ω_2 into those that include q_2 ($q_2 \leq q_2''$) and those that do not. In the first case, since from (8.27), (8.21), and (8.17), $\Omega_{q_1\mathcal{P}_2tr_1q_2''} \subseteq \Omega_{m(i+1)}$, and since each element q_2' of $\Omega_{q_1\mathcal{P}_2tr_1q_2''}$ satisfies $q_2 \leq q_2'$,

$$\sum_{q_2'\in\Omega_{m(i+1)}|q_2\leq q_2'} P_{q_1\mathcal{P}_2tr_1q_2''}[q_2'] = 1;\tag{8.43}$$

in the second case the same sum gives $P_{q_2}^{\mathcal{O}_{q_1\mathcal{P}_2tr_1q_2''}}$. Consider the second summand of Expression (8.42), and observe that, from (8.27), (8.21), and the definition of $\delta_{\mathcal{P}_2}$, $q_2'\delta \in \Omega_{m(i+1)}$, $q_2 \leq q_2'$, and $P_2[q_2'] > 0$ iff $q_2' \in \Omega_2$, $q_2 \leq q_2'$, and $P_2[q_2'] > 0$. Finally, consider the third summand of Expression (8.42), and observe that all the states of Ω_2 end with δ, and, from (8.27) and (8.21), $q_2'\delta \in \Omega_{m(i+1)}$, $q_2 \leq q_2'$, and $P_2[q_2'\delta] > 0$ iff $q_2'\delta \in \Omega_2$, $q_2 \leq q_2'\delta$, $P_2[q_2'\delta] > 0$. By combining the observations above, Expression (8.42) can be rewritten into

$$\begin{aligned}&\sum_{(q_1,\mathcal{P}_2)\in Active(i),N(q_1)}\ \sum_{tr_1\in\Omega^{q_1}_{H_1}} w_i(q_1,\mathcal{P}_2)P^{q_1}_{H_1}[tr_1]\\&\qquad\left(\sum_{q_2''\in\Omega_2|q_2\leq q_2''} P_2[q_2''] + \sum_{q_2''\in\Omega_2|q_2''<q_2} P_2[q_2'']P_{q_2}^{\mathcal{O}_{q_1\mathcal{P}_2tr_1q_2''}}\right)\\&+\sum_{(q_1,\mathcal{P}_2)\in Active(i),N(q_1)}\ \sum_{q_2''\in\Omega_2|q_2\leq q_2''} w_i(q_1,\mathcal{P}_2)P^{q_1}_{H_1}[\delta]P_2[q_2'']\\&+\sum_{(q_1\delta,\mathcal{P}_2)\in Active(i)}\ \sum_{q_2''\in\Omega_2|q_2\leq q_2''} w_i(q_1\delta,\mathcal{P}_2)P_2[q_2''].\end{aligned}\tag{8.44}$$

By regrouping expressions and simplifying, we obtain

$$\begin{aligned}&\sum_{(q_1,\mathcal{P}_2)\in Active(i),N(q_1)}\ \sum_{tr_1\in\Omega^{q_1}_{H_1}}\ \sum_{q_2''\in\Omega_2|q_2\leq q_2''} w_i(q_1,\mathcal{P}_2)P^{q_1}_{H_1}[tr_1]P_2[q_2'']P_{q_2}^{\mathcal{O}_{q_1\mathcal{P}_2tr_1q_2''}}\\&+\sum_{(q_1,\mathcal{P}_2)\in Active(i)}\ \sum_{q_2''\in\Omega_2|q_2\leq q_2''} w_i(q_1,\mathcal{P}_2)P_2[q_2''].\end{aligned}\tag{8.45}$$

Finally, from Equation (8.30), Expression (8.45) can be rewritten into

$$\sum_{(q_1,\mathcal{P}_2)\in Active(i),N(q_1)} \sum_{tr_1\in\Omega_{H_1}^{q_1}} \sum_{q_2''\in\Omega_2|q_2\leq q_2''} W_{q_1\mathcal{P}_2tr_1q_2''}(q_2) \tag{8.46}$$
$$+ \sum_{(q_1,\mathcal{P}_2)\in Active(i)} \sum_{q_2''\in\Omega_2|q_2\leq q_2''} w_i(q_1,\mathcal{P}_2)P_2[q_2''].$$

We now analyze the second summand of Expression (8.46), and we show by induction on i that it is 0 if $i = 0$ and $q_2 \neq s_2$, it is 1 if $i = 0$ and $q_2 = s_2$, and it is

$$\sum_{j<i} \sum_{(q_1,\mathcal{P}_2)\in Active(j)} \sum_{tr_1\in\Omega_{H_1}^{q_1}} \sum_{q_2''\in\Omega_2|q_2''<q_2} W_{q_1\mathcal{P}_2tr_1q_2''}(q_2) \tag{8.47}$$

otherwise. For $i = 0$ the result is trivial. Otherwise, from Equation (8.29),

$$\sum_{(q_1,\mathcal{P}_2)\in Active(i+1)} \sum_{q_2''\in\Omega_2|q_2\leq q_2''} w_{i+1}(q_1,\mathcal{P}_2)P_2[q_2''] \tag{8.48}$$

can be rewritten into

$$\sum_{(q_1,\mathcal{P}_2)\in Active(i+1)} \sum_{(q_1',\mathcal{P}_2')\in Active(i)} \sum_{q_2''\in\Omega_2|q_2\leq q_2''} w_i(q_1',\mathcal{P}_2')w_{q_1'\mathcal{P}_2'}(q_1,\mathcal{P}_2)P_2[q_2'']. \tag{8.49}$$

From the definition of $w_{q_1'\mathcal{P}_2'}$ (Equations (8.23) and (8.25)), Expression (8.49) can be rewritten into

$$\sum_{(q_1,\mathcal{P}_2)\in Active(i+1)} \sum_{(q_1',\mathcal{P}_2')\in Active(i),N(q_1')} \sum_{tr_1'\in\Omega_{H_1}^{q_1'}} \sum_{q_2''\in\Omega_2|q_2\leq q_2''} \tag{8.50}$$
$$w_i(q_1',\mathcal{P}_2')P_{H_1}^{q_1'}[tr_1']w_{q_1'\mathcal{P}_2'tr_1'}(q_1,\mathcal{P}_2))P_2[q_2'']$$
$$+ \sum_{(q_1\delta,\mathcal{P}_2)\in Active(i+1)} \sum_{(q_1',\mathcal{P}_2')\in Active(i),N(q_1')} \sum_{q_2''\in\Omega_2|q_2\leq q_2''}$$
$$w_i(q_1',\mathcal{P}_2')P_{H_1}^{q_1'}[\delta]w_{\delta q_1'\mathcal{P}_2'}(q_1\delta,\mathcal{P}_2)P_2[q_2'']$$
$$+ \sum_{(q_1'\delta,\mathcal{P}_2')\in Active(i)} \sum_{q_2''\in\Omega_2'|q_2\leq q_2''} w_i(q_1'\delta,\mathcal{P}_2')P_2[q_2''].$$

Observe that in the first summand of (8.50)

$$\sum_{(q_1,\mathcal{P}_2)\in Active(i+1)} \sum_{q_2''\in\Omega_2|q_2\leq q_2''} w_{q_1'\mathcal{P}_2'tr_1'}(q_1,\mathcal{P}_2)P_2[q_2'']$$
$$= \sum_{\mathcal{P}_2|\exists q_1,(q_1,\mathcal{P}_2)\in Active(i+1)} \sum_{q_2''\in\Omega_2|q_2\leq q_2''} P^S_{q_1'\mathcal{P}_2'tr_1'}[\mathcal{P}_2]P_2[q_2'']$$
$$= \sum_{q_2'''\in\Omega_2'} \sum_{q_2''\in\Omega_{q_1'\mathcal{P}_2'tr_1'}|q_2\leq q_2''} P_{q_1'\mathcal{P}_2'tr_1'q_2'''}[q_2''],$$

where the first step follows from the fact that $w_{q_1'\mathcal{P}_1'tr_1'q_2'''}$ is a weight function, and the second step follows from (8.17), (8.15) and the fact that $\Omega_{q_1'\mathcal{P}_2'tr_1'}$ is the set of probability space $\mathcal{P}_2$ such that there is a state q_1 where $(q_1, \mathcal{P}_2) \in \mathit{Active}(i+1)$ (cf. the definition of *Active* and observe that $|q_1| = i+1$). For the second summand of (8.50), observe that for each pair $(q_1'\delta, \mathcal{P}_2)$ of $\mathit{Active}(i+1)$, if $P_{H_1}^{q_1'}[\delta] > 0$, then there is exactly one pair $(q_1, \mathcal{P}_2')$ of $\mathit{Active}(i)$ such that $w_{\delta q_1'\mathcal{P}_2'}(q_1'\delta, \mathcal{P}_2) > 0$. In particular, $q_1 = q_1'$, $\mathcal{P}_2 = \delta_{\mathcal{P}_2'}$, and $w_{\delta q_1'\mathcal{P}_2'}(q_1'\delta, \mathcal{P}_2) = 1$. Conversely, for each pair $(q_1', \mathcal{P}_2')$ of $\mathit{Active}(i)$ such that $P_{H_1}^{q_1'}[\delta] > 0$, the pair $(q_1'\delta, \mathcal{P}_2)$ is in $\mathit{Active}(i+1)$ and $w_{\delta q_1'\mathcal{P}_2'}(q_1'\delta, \mathcal{P}_2) = 1$. Thus, the term $w_{\delta q_1'\mathcal{P}_2'}(q_1'\delta, \mathcal{P}_2)$ and the sum $\sum_{(q_1'\delta, \mathcal{P}_2) \in \mathit{Active}(i+1)}$ can be removed from the second summand of (8.50). Thus, by applying the observations above to (8.50), we obtain

$$\begin{aligned}
&\sum_{(q_1',\mathcal{P}_2') \in \mathit{Active}(i), N(q_1')} \; \sum_{tr_1' \in \Omega_{H_1}^{q_1'}} \; \sum_{q_2''' \in \Omega_2'} \; \sum_{q_2'' \in \Omega_{q_1'\mathcal{P}_2'tr_1'q_2'''} | q_2 \leq q_2''} \\
&\quad w_i(q_1', \mathcal{P}_2') P_{H_1}^{q_1'}[tr_1'] P_2'[q_2'''] P_{q_1'\mathcal{P}_2'tr_1'q_2'''}[q_2''] \\
&+ \sum_{(q_1',\mathcal{P}_2') \in \mathit{Active}(i), N(q_1')} \; \sum_{q_2''' \in \Omega_2' | q_2 \leq q_2''} w_i(q_1', \mathcal{P}_2') P_{H_1}^{q_1'}[\delta] P_2'[q_2'''] \\
&+ \sum_{(q_1'\delta,\mathcal{P}_2') \in \mathit{Active}(i)} \; \sum_{q_2''' \in \Omega_2' | q_2 \leq q_2'''} w_i(q_1'\delta, \mathcal{P}_2') P_2'[q_2'''].
\end{aligned} \tag{8.51}$$

Consider the first summand of Expression (8.51). If $q_2 \leq q_2'''$, then

$$\sum_{q_2'' \in \Omega_{q_1'\mathcal{P}_2'tr_1'q_2'''} | q_2 \leq q_2''} P_{q_1'\mathcal{P}_2'tr_1'q_2'''}[q_2''] = 1; \tag{8.52}$$

If $q_2''' \leq q_2$, then

$$\sum_{q_2'' \in \Omega_{q_1'\mathcal{P}_2'tr_1'q_2'''} | q_2 \leq q_2''} P_{q_1'\mathcal{P}_2'tr_1'q_2'''}[q_2''] = P_{q_2}^{\mathcal{O}_{q_1'\mathcal{P}_2'tr_1'q_2'''}}. \tag{8.53}$$

Thus, from Equations (8.52) and (8.53), Expression (8.51) can be rewritten into

$$\begin{aligned}
&\sum_{(q_1',\mathcal{P}_2') \in \mathit{Active}(i), N(q_1')} \; \sum_{tr_1' \in \Omega_{H_1}^{q_1'}} w_i(q_1', \mathcal{P}_2') P_{H_1}^{q_1'}[tr_1'] \\
&\quad \left(\sum_{q_2''' \in \Omega_2' | q_2 \leq q_2'''} P_2'[q_2'''] + \sum_{q_2''' \in \Omega_2' | q_2''' < q_2} P_2'[q_2'''] P_{q_2}^{\mathcal{O}_{q_1'\mathcal{P}_2'tr_1'q_2'''}} \right) \\
&+ \sum_{(q_1',\mathcal{P}_2') \in \mathit{Active}(i), N(q_1')} \; \sum_{q_2''' \in \Omega_2' | q_2 \leq q_2''} w_i(q_1', \mathcal{P}_2') P_{H_1}^{q_1'}[\delta] P_2'[q_2'''] \\
&+ \sum_{(q_1'\delta,\mathcal{P}_2') \in \mathit{Active}(i)} \; \sum_{q_2''' \in \Omega_2' | q_2 \leq q_2'''} w_i(q_1'\delta, \mathcal{P}_2') P_2'[q_2'''].
\end{aligned} \tag{8.54}$$

By regrouping the subexpressions in (8.54), we obtain

$$\sum_{(q_1',\mathcal{P}_2')\in Active(i),N(q_1')} \sum_{tr_1'\in\Omega_{H_1}^{q_1'}} \sum_{q_2'''\in\Omega_2'|q_2'''<q_2} w_i(q_1',\mathcal{P}_2')P_{H_1}^{q_1'}[tr_1']P_2'[q_2''']P_{q_2}^{\mathcal{O}_{q_1'\mathcal{P}_2'tr_1'q_2'''}} \tag{8.55}$$
$$+ \sum_{(q_1',\mathcal{P}_2')\in Active(i)} \sum_{q_2'''\in\Omega_2'|q_2\leq q_2'''} w_i(q_1',\mathcal{P}_2')P_2'[q_2'''].$$

From Equation (8.30), Expression (8.55) can be rewritten into

$$\sum_{(q_1',\mathcal{P}_2')\in Active(i),N(q_1')} \sum_{tr_1'\in\Omega_{H_1}^{q_1'}} \sum_{q_2'''\in\Omega_2'|q_2'''<q_2} W_{q_1'\mathcal{P}_2'tr_1'q_2'''}(q_2) \tag{8.56}$$
$$+ \sum_{(q_1',\mathcal{P}_2')\in Active(i)} \sum_{q_2'''\in\Omega_2'|q_2\leq q_2'''} w_i(q_1',\mathcal{P}_2')P_2'[q_2'''].$$

The induction hypothesis is now sufficient to conclude the validity of (8.47). From an alternative characterization of the set $\{q_2''\in\Omega_2 \mid q_2''<q_2\}$ in Expressions (8.47) and (8.45), and by combining (8.45) and (8.47), we obtain

$$\sum_{q_2'\in\Omega_{m(i+1)}|q_2\leq q_2'} P_{m(i+1)}[q_2'] \tag{8.57}$$
$$= \sum_{j\leq i} \sum_{(q_1,\mathcal{P}_2)\in Active(j)} \sum_{tr_1\in\Omega_{H_1}^{q_1}} \sum_{q_2''\in\Omega_2|q_2''\neq q_2,q_2''\in reach(q_1,\mathcal{P}_2,tr_1,q_2)} W_{q_1\mathcal{P}_2tr_1q_2''}(q_2).$$

Observe that the right expression of (8.57) contains a subset of the terms of the right expression of Equation (8.32). This is enough to conclude the validity of (8.35).

2. For each i, $m(i)\leq m(i+1)$.

   This result follows directly from Equation (8.57). In fact, for each state q_2 of H_2, Expression (8.57) for $m(i+1)$ contains a subset of the terms of the Expression (8.57) for $m(i)$.

3. For each state q of H_2, $\lim_{i\to\infty}\sum_{q'\in\Omega_i|q\leq q'}P_i[q'] = P_H[C_q]$.

   This result follows directly from Expression (8.57). In fact, as $i\to\infty$, the right expression of (8.57) converges to the right expression of (8.32).

4. For each i, $m(i)=\sum_{\mathcal{P}\in S(i)}P_{S(i)}[\mathcal{P}]\mathcal{P}$.

   For $i=0$ the result is trivial. For $i>0$, from Equation (8.27), $m(i+1)$ is rewritten into.

$$\sum_{(q_1,\mathcal{P}_2)\in Active(i)} w_i(q_1,\mathcal{P}_2)\mathcal{P}_{q_1\mathcal{P}_2}. \tag{8.58}$$

From Equation (8.21), Expression (8.58) can be rewritten into

$$\sum_{(q_1,\mathcal{P}_2)\in Active(i)} w_i(q_1,\mathcal{P}_2)\left(\sum_{tr_1\in\Omega_{H_1}^{q_1}} P_{H_1}^{q_1}[tr_1]\mathcal{P}_{q_1\mathcal{P}_2tr_2} + P_{H_1}^{q_1}[\delta]\delta_{\mathcal{P}_2}\right). \tag{8.59}$$

From Equation (8.17) applied to $\mathcal{P}_{q_1\mathcal{P}_2 tr_2}$ and Equations (8.15) and (8.19) applied to $P^{q_1}_{H_1}[\delta]\delta_{\mathcal{P}_2}$, Expression (8.59) can be rewritten into

$$\sum_{(q_1,\mathcal{P}_2)\in \mathit{Active}(i)} w_i(q_1,\mathcal{P}_2)\left(\sum_{tr_1\in\Omega^{q_1}_{H_1}} P^{q_1}_{H_1}[tr_1]\left(\sum_{\mathcal{P}\in\Omega^S_{q_1\mathcal{P}_2 tr_1}} P^S_{q_1\mathcal{P}_2 tr_1}[\mathcal{P}]\mathcal{P}\right)+ \right. \qquad (8.60)$$
$$\left. P^{q_1}_{H_1}[\delta]\sum_{\mathcal{P}\in\Omega_{\delta_{\mathcal{P}_2}}} P_{\delta_{\mathcal{P}_2}}[\mathcal{P}]\mathcal{P}\right).$$

From Equation (8.22), Expression (8.60) can be rewritten into

$$\sum_{(q_1,\mathcal{P}_2)\in \mathit{Active}(i)} w_i(q_1,\mathcal{P}_2)\left(\sum_{\mathcal{P}\in\Omega^S_{q_1\mathcal{P}_2}} P^S_{q_1\mathcal{P}_2}[\mathcal{P}]\mathcal{P}\right). \qquad (8.61)$$

Finally, from Equation (8.28), Expression (8.61) can be rewritten into

$$\sum_{\mathcal{P}\in\Omega_{S(i+1)}} P_{S(i+1)}[\mathcal{P}]\mathcal{P}, \qquad (8.62)$$

which is what we needed to show.

5. For each i, $\mathit{fringe}(H_1,i)\sqsubseteq_{\mathcal{R}} S(i)$ via w_i.

   For $i=0$ the result is trivial. By applying the definitions of a fringe and of $\mathit{fringe}(H_1,i{+}1)$,

$$\begin{aligned}&\mathit{fringe}(H_1,i+1)\\ &\quad=\sum_{q_1\in \mathit{states}(H_2)||q_2|=i\text{or } q_2=q_2'\delta,|q_2|<i} P_{H_1}[C_{q_1}]\mathcal{P}_{q_1}\\ &\quad=\sum_{(q_1,\mathcal{P}_2)\in \mathit{Active}(i)} w_i(q_1,\mathcal{P}_2)\mathcal{P}_{q_1}.\end{aligned}$$

   From (8.28),

$$S(i+1)=\sum_{(q_1,\mathcal{P}_2)\in \mathit{Active}(i)} w_i(q_1,\mathcal{P}_2)\mathcal{P}^S_{q_1\mathcal{P}_2}.$$

   Since for each pair $(q_1,\mathcal{P}_2)$ of $\mathit{Active}(i)$, $\mathcal{P}_{q_1}\sqsubseteq_{\mathcal{R}}\mathcal{P}^S_{q_1\mathcal{P}_2}$ via $w_{q_1,\mathcal{P}_2}$, from Lemma 8.2.1,

$$\sum_{(q_1,\mathcal{P}_2)\in \mathit{Active}(i)} w_i(q_1,\mathcal{P}_2)\mathcal{P}_{q_1}\sqsubseteq_{\mathcal{R}}\sum_{(q_1,\mathcal{P}_2)\in \mathit{Active}(i)} w_i(q_1,\mathcal{P}_2)\mathcal{P}^S_{q_1\mathcal{P}_2}$$

   via $\sum_{(q_1,\mathcal{P}_2)\in \mathit{Active}(i)} w_i(q_1,\mathcal{P}_2)w_{q_1\mathcal{P}_2}$, which is w_{i+1}. ■

6. For each i, each $q_1 \in \mathit{fringe}(H_1, i)$, and each $q_2 \in \mathit{states}(H_2)$, if $W_i(q_1, q_2') = 0$ for each prefix or extension q_2' of q_2, then, for each extension q_1' of q_1 such that $q_1' \in \mathit{fringe}(H_1, i{+}1)$ and each prefix or extension q_2' of q_2, $W_{i+1}(q_1', q_2') = 0$.

   Suppose by contradiction that there is an extension q_1' of q_1 such that $q_1' \in \mathit{fringe}(H_1, i{+}1)$ and a prefix or extension q_2' of q_2 such that $W_{i+1}(q_1', q_2') > 0$. From the definition of W_i and Equation (8.29),

$$W_{i+1}(q_1, q_2') = \sum_{\mathcal{P}} \sum_{(\bar{q}_1, \mathcal{P}_2) \in \mathit{Active}(i)} w_i(\bar{q}_1, \mathcal{P}_2) w_{\bar{q}_1, \mathcal{P}_2}(q_1, \mathcal{P}) P[q_2']. \tag{8.63}$$

   Since $W_i(q_1, q_2') > 0$, then there is at least one probability space $\mathcal{P}$ and one pair $(\bar{q}_1, \mathcal{P}_2) \in \mathit{Active}(i)$ such that $w_i(\bar{q}_1, \mathcal{P}_2) > 0$, $w_{\bar{q}_1, \mathcal{P}_2}(q_1, \mathcal{P}) > 0$, and $P[q_2'] > 0$. Then there is at least one prefix q_2'' of q_2' such that $P_2[q_2''] > 0$, which means that $W_i(\bar{q}_1, q_2'') > 0$. However, this is a contradiction since q_2'' is either a prefix or a suffix of q_2.

The execution correspondence theorem can be stated and proved similarly for weak and strong probabilistic simulations. The proofs are simpler than the proof presented in this section, and thus we omit them from this thesis.

8.6.4 Transitivity of Probabilistic Forward Simulations

Now we have enough machinery to prove that probabilistic forward simulations are transitive, i.e., if $M_1 \sqsubseteq_{FS} M_2$ and $M_2 \sqsubseteq_{FS} M_3$, then $M_1 \sqsubseteq_{FS} M_3$. We start by proving a lemma.

Lemma 8.6.2 *Let (H_1, H_2, m, S) be an execution correspondence structure via the probabilistic forward simulation $\mathcal{R}$, and suppose that H_1 represents a weak combined transition $s \stackrel{a}{\Longrightarrow}_{\rm C} \mathcal{P}_1$. Then H_2 represents a weak combined transition $s' \stackrel{a}{\Longrightarrow}_{\rm C} \mathcal{P}_2$ and there is a probability space $\mathcal{P}_2^S$ such that*

1. *$\mathcal{P}_1 \sqsubseteq_{\mathcal{R}} \mathcal{P}_2^S$ and*

2. *$\mathcal{P}_2 = \sum_{\mathcal{P} \in \Omega_2^S} P_2^S[\mathcal{P}]\mathcal{P}$.*

Proof. Let w_i be the weight functions for $\mathit{fringe}(H_1, i) \sqsubseteq_{\mathcal{R}} S(i)$. Let $\mathcal{P}_1'$ be $\delta\text{-}\mathit{strip}(\mathcal{P}_{H_1})$, $\mathcal{P}_2'$ be $\delta\text{-}\mathit{strip}(\mathcal{P}_{H_2})$, and let

$$\mathcal{P}_{2,S}' \triangleq \sum_{\alpha\delta \in \Omega_{H_1}} \sum_{\mathcal{P} \mid w_{|\alpha|+1}(\alpha\delta, \mathcal{P}) > 0} w_{|\alpha|+1}(\alpha\delta, \mathcal{P})\mathcal{P}. \tag{8.64}$$

For each $\alpha\delta \in \Omega_{H_1}$ and each $\mathcal{P} \in \mathit{Probs}(\mathit{extstates}(H_2))$, let $w(\alpha\delta, \mathcal{P}) \triangleq w_{|\alpha|+1}(\alpha\delta, \mathcal{P})$.

We show that w is a weight function from $\mathcal{P}_1'$ to $\mathcal{P}_{2,S}'$ and that $\mathcal{P}_{2,S}'$ is well defined. This implies that $\mathcal{P}_1' \sqsubseteq_{\mathcal{R}} \mathcal{P}_{2,S}'$. Then we show that for each element $\alpha\delta$ of Ω_{H_2}, $\sum_{\mathcal{P} \in \Omega_{2,S}'} P_{2,S}'[\mathcal{P}]P[\alpha\delta] = P_{H_2}[C_{\alpha\delta}]$. Since all the elements of the probability spaces of $\Omega_{2,S}'$ end with δ, we obtain that $\mathcal{P}_2'$ is well defined and that $\mathcal{P}_2' = \sum_{\mathcal{P} \in \Omega_{2,S}'} P_{2,S}'[\mathcal{P}]\mathcal{P}$. Then the lemma is proved by defining $\mathcal{P}_1$ to be $\mathit{lstate}(\mathcal{P}_1')$, $\mathcal{P}_2$ to be $\mathit{lstate}(\mathcal{P}_2')$, and $\mathcal{P}_{2,S}$ to be $\mathit{lstate}(\mathcal{P}_{2,S}')$.

To show that w is a weight function we have to verify the three conditions of the definition of a weight function. If $w(\alpha\delta, \mathcal{P}) > 0$, then, from the definition of w, $w_{|\alpha|+1}(\alpha\delta, \mathcal{P}) > 0$.

Since $w_{|\alpha|+1}$ is a weight function, then $\alpha\delta \mathrel{\mathcal{R}} \mathcal{P}$. Let $\mathcal{P} \in \Omega'_{2,S}$. Then $\sum_{\alpha\delta\in\Omega_{H_1}} w(\alpha\delta,\mathcal{P}) = \sum_{\alpha\delta\in\Omega_{H_1}} w_{|\alpha|+1}(\alpha\delta,\mathcal{P})$, which is $P'_{2,S}[\mathcal{P}]$ by definition of $\mathcal{P}'_{2,S}$. Consider now an element $\alpha\delta$ of Ω_{H_1}. Then, $\sum_{\mathcal{P}\in\Omega'_{2,S}} w(\alpha\delta,\mathcal{P}) = \sum_{\mathcal{P}\in\Omega'_{2,S}} w_{|\alpha|+1}(\alpha\delta,\mathcal{P})$. Since $w_{|\alpha|+1}$ is a weight function, then the sum above gives $P_{H_1}[C_{\alpha\delta}] = P'_1[\alpha\delta]$. To show that $\mathcal{P}'_{2,S}$ is well defined we need to show that $\sum_{\alpha\delta\in\Omega_{H_1}} \sum_{\mathcal{P}|w_{|\alpha|+1}(\alpha\delta,\mathcal{P})>0} w_{|\alpha|+1}(\alpha\delta,\mathcal{P}) = 1$. This follows immediately from the fact that w is a weight function and that, since H_1 represents a weak combined transition, $\sum_{\alpha\delta\in\Omega_{H_1}} P'_1[\alpha\delta] = 1$.

We are left to show that for each element $\alpha\delta$ of Ω_{H_2}, $\sum_{\mathcal{P}\in\Omega'_{2,S}} P'_{2,S}[\mathcal{P}]P[\alpha\delta] = P_{H_2}[C_{\alpha\delta}]$. Observe that for each element $\alpha\delta$ of Ω_{H_1}, if $i \leq |\alpha|$ then $w_i(\alpha\delta,\mathcal{P})$ is undefined for each $\mathcal{P}$, and if $i > |\alpha|$, then for each $j \geq i$ and each $\mathcal{P}$, $w_i(\alpha\delta,\mathcal{P})$ is defined iff $w_j(\alpha\delta,\mathcal{P})$ is defined, and if $w_i(\alpha\delta,\mathcal{P})$ is defined then $w_i(\alpha\delta,\mathcal{P}) = w_j(\alpha\delta,\mathcal{P})$. Thus, if we extend each w_i by setting it to 0 whenever it is not defined, then, for each $\alpha\delta \in \Omega_{H_2}$,

$$\sum_{\mathcal{P}\in\Omega'_{2,S}} P'_{2,S}[\mathcal{P}]P[\alpha\delta] = \sum_{\mathcal{P}\in\Omega'_{2,S}} \left(\lim_{i\to\infty} \sum_{\alpha\delta\in\Omega_{H_1}} w_i(\alpha\delta,\mathcal{P}) \right) P[\alpha\delta]. \tag{8.65}$$

Since for each i, w_i is a weight function, and since from the definition of $\mathcal{P}'_{2,S}$ each element $\mathcal{P}$ for which $w_i(\alpha\delta,\mathcal{P}) > 0$ is in $\Omega'_{2,S}$, then we derive

$$\sum_{\mathcal{P}\in\Omega'_{2,S}} P'_{2,S}[\mathcal{P}]P[\alpha\delta] = \sum_{\mathcal{P}\in\Omega'_{2,S}} \left(\lim_{i\to\infty} P_{S(i)}[\mathcal{P}] \right) P[\alpha\delta]. \tag{8.66}$$

By exchanging the limit with the sum and by using Condition 3 of the definition of an execution correspondence structure, the equation above can be rewritten into

$$\sum_{\mathcal{P}\in\Omega'_{2,S}} P'_{2,S}[\mathcal{P}]P[\alpha\delta] = \lim_{i\to\infty} m(i)[\alpha\delta], \tag{8.67}$$

which gives the desired result after using Condition 2 of the definition of an execution correspondence structure. ■

Proposition 8.6.3 *Probabilistic forward simulations are transitive.*

Proof. Let $\mathcal{R}_1$ be a probabilistic forward simulation from M_1 to M_2, and let $\mathcal{R}_2$ be a probabilistic forward simulation from M_2 to M_3. Define $\mathcal{R}$ so that $s_1 \mathrel{\mathcal{R}} \mathcal{P}_3$ iff there is a probability space $\mathcal{P}_2$, and a probability space $\mathcal{P}_3^S$, such that

1. $s_1 \mathrel{\mathcal{R}_1} \mathcal{P}_2$,

2. $\mathcal{P}_2 \sqsubseteq_{\mathcal{R}_2} \mathcal{P}_3^S$, and

3. $\mathcal{P}_3 = \sum_{\mathcal{P}\in\Omega_3^S} P_3^S[\mathcal{P}]\mathcal{P}$.

We need to show that $\mathcal{R}$ is a probabilistic forward simulation from M_1 to M_3. For this purpose, let $s_1 \mathrel{\mathcal{R}} \mathcal{P}_3$, and let $\mathcal{P}_2$ and $\mathcal{P}_3^S$ satisfy the three conditions above. Let $s_1 \xrightarrow{a} \mathcal{P}_1$. Let M'_2 be obtained from M_2 by introducing a new state s'_2 and by adding a transition $s'_2 \xrightarrow{\tau} \mathcal{P}_2$, where τ is an internal action; similarly, let M'_3 be obtained from M_3 by introducing a new state s'_3 and by adding a transition $s'_3 \xrightarrow{\tau} \mathcal{P}_3$, where τ is an internal action. Let $\mathcal{R}'_1$ be obtained from $\mathcal{R}_1$ by adding the pair $(s_1, \mathcal{D}(s_2'))$, and let $\mathcal{R}_2'$ be obtained from $\mathcal{R}_2$ by adding the pair $(s_2', \mathcal{D}(s_3'))$. Observe that $\mathcal{R}_1'$ is a probabilistic forward simulation from M_1 to M_2' and that $\mathcal{R}_2'$ is a probabilistic forward simulation from M_2' to M_3'.

We want to find two probability spaces $\mathcal{P}_3'$ and $\mathcal{P}_{3,S}'$ such that $s_3' \overset{a}{\Longrightarrow}_C \mathcal{P}_3'$, $\mathcal{P}_1' \sqsubseteq_{\mathcal{R}} \mathcal{P}_{3,S}'$, and $\mathcal{P}_3' = \sum_{\mathcal{P} \in \Omega_{3,S}'} P_{3,S}'[\mathcal{P}]\mathcal{P}$. From the definition of a weak transition, this is sufficient to show that for each state s of $\mathcal{P}_3$ there is a weak combined transition $s \overset{a}{\Longrightarrow}_C \mathcal{P}_s$ of M_3 such that $\mathcal{P}_3' = \sum_{s \in \Omega_3} P_3[s]\mathcal{P}_s$.

Since $\mathcal{R}_1'$ is a probabilistic forward simulation, there is a weak combined transition $s_2' \overset{a}{\Longrightarrow}_C \mathcal{P}_2'$ of M_2' and a probability space $\mathcal{P}_{2,S}'$ such that

$$\mathcal{P}_2' = \sum_{\mathcal{P} \in \Omega_{2,S}'} P_{2,S}'[\mathcal{P}]\mathcal{P} \quad \text{and} \quad \mathcal{P}_1' \sqsubseteq_{\mathcal{R}_1} \mathcal{P}_{2,S}'. \tag{8.68}$$

Let H_2 be the probabilistic execution fragment of M_2' that represents the weak combined transition $s_2' \overset{a}{\Longrightarrow}_C \mathcal{P}_2'$. Then, by definition of H_2, $\mathcal{P}_2' = \mathit{lstate}(\delta\text{-}\mathit{strip}(\mathcal{P}_{H_2}))$ (cf. Section 4.2.7).

From the Execution Correspondence Theorem there is an execution correspondence structure (H_2, H_3, m, S), where H_3 is a probabilistic execution fragment of M_3' that starts from s_3'. From Lemma 8.6.2, H_3 represents a weak combined transition $s_3' \overset{a}{\Longrightarrow}_C \mathcal{P}_3''$ for same probability space $\mathcal{P}_3''$. Moreover, there is a probability space $\mathcal{P}_{3,S}''$ such that

$$\mathcal{P}_3'' = \sum_{\mathcal{P} \in \Omega_{3,S}''} P_{3,S}''[\mathcal{P}]\mathcal{P} \quad \text{and} \quad \mathcal{P}_2' \sqsubseteq_{\mathcal{R}_2} \mathcal{P}_{3,S}''. \tag{8.69}$$

Let w_2 be the weight function for $\mathcal{P}_2' \sqsubseteq_{\mathcal{R}_2} \mathcal{P}_{3,S}''$. For each probability space $\mathcal{P}$ of $\Omega_{2,S}'$, let $w_{\mathcal{P}} : \mathit{states}(M_2) \times \mathit{Probs}(\mathit{states}(M_3)) \to [0,1]$ be a function that is non-zero only in the set $\Omega \times \Omega_{3,S}''$ and such that for each pair $(s, \mathcal{P}')$ of $\Omega \times \Omega_{3,S}''$,

$$w_{\mathcal{P}}(s, \mathcal{P}') = \frac{P[s]w_2(s, \mathcal{P}')}{P_2'[s]}. \tag{8.70}$$

Also, for each probability space $\mathcal{P}$ of $\Omega_{2,S}'$, let

$$\mathcal{P}_{3,S}^{\mathcal{P}} \triangleq \sum_{s \in \Omega} \sum_{\mathcal{P}' \in \Omega_{3,S}''} w_{\mathcal{P}}(s, \mathcal{P}')\mathcal{D}(\mathcal{P}'), \tag{8.71}$$

and let

$$\mathcal{P}_3^{\mathcal{P}} \triangleq \sum_{\mathcal{P}' \in \Omega_{3,S}^{\mathcal{P}}} P_{3,S}^{\mathcal{P}}[\mathcal{P}']\mathcal{P}'. \tag{8.72}$$

Let $\mathcal{P}_{3,S}'$ be the discrete probability space where $\Omega_{3,S}' = \{\mathcal{P}_3^{\mathcal{P}} \mid \mathcal{P} \in \Omega_{2,S}\}$, and for each element $\mathcal{P}_3^{\mathcal{P}}$ of $\Omega_{3,S}'$, $P_{3,S}'[\mathcal{P}_3^{\mathcal{P}}] = \sum_{\mathcal{P}' \in \Omega_{2,S}' | \mathcal{P}_3^{\mathcal{P}} = \mathcal{P}_3^{\mathcal{P}'}} P_{2,S}'[\mathcal{P}']$. Then, the following properties are true.

1. For each probability space $\mathcal{P}$ of $\Omega_{2,S}'$, $w_{\mathcal{P}}$ is a weight function from $\mathcal{P}$ to $\mathcal{P}_{3,S}^{\mathcal{P}}$.

   We verify separately each one of the conditions that a weight function must satisfy.

(a) For each $s \in \mathit{states}(M_2)$, $P[s] = \sum_{\mathcal{P}' \in \mathit{Probs}(\mathit{states}(M_3))} w_{\mathcal{P}}(s, \mathcal{P}')$.

From the definition of $w_{\mathcal{P}}$, the right expression above can be rewritten into

$$\sum_{\mathcal{P}' \in \mathit{Probs}(\mathit{states}(M_3))} \frac{P[s] w_2(s, \mathcal{P}')}{P_2'[s]}. \tag{8.73}$$

Since w_2 is a weight function, $\sum_{\mathcal{P}' \in \mathit{Probs}(\mathit{states}(M_3))} w_2(s, \mathcal{P}') = P_2'[s]$, and thus Expression 8.73 becomes $P[s]$.

(b) For each $\mathcal{P}' \in \mathit{Probs}(\mathit{states}(M_3))$, $\sum_{s \in \mathit{states}(M_2)} w_{\mathcal{P}}(s, \mathcal{P}') = P_{3,S}^{\mathcal{P}}[\mathcal{P}']$.

From Equation (8.71), $P_{3,S}^{\mathcal{P}}[\mathcal{P}'] = \sum_{s \in \Omega} w_{\mathcal{P}}(s, \mathcal{P}')$. Since $w_{\mathcal{P}}$ is non-zero only when the first argument is in Ω, $P_{3,S}^{\mathcal{P}}[\mathcal{P}'] = \sum_{s \in \mathit{states}(M_2)} w_{\mathcal{P}}(s, \mathcal{P}')$.

(c) For each $(s, \mathcal{P}') \in \mathit{states}(M_2) \times \mathit{Probs}(\mathit{states}(M_3))$, if $w_{\mathcal{P}}(s, \mathcal{P}') > 0$ then $s \mathrel{\mathcal{R}_2} \mathcal{P}'$.

If $w_{\mathcal{P}}(s, \mathcal{P}') > 0$, then, from Equation (8.70), $w_2(s, \mathcal{P}') > 0$. Since w_2 is a weight function, then $s \mathrel{\mathcal{R}_2} \mathcal{P}'$.

2. $\sum_{\mathcal{P} \in \Omega_{3,S}'} P_{3,S}'[\mathcal{P}]\mathcal{P} = \mathcal{P}_3''$.

From the definition of $\mathcal{P}_{3,S}'$, Equation (8.72), Equation (8.71), and Equation (8.70), $\sum_{\mathcal{P} \in \Omega_{3,S}'} P_{3,S}'[\mathcal{P}]\mathcal{P}$ can be rewritten into

$$\sum_{\mathcal{P} \in \Omega_{2,S}'} \sum_{\mathcal{P}' \in \Omega_{3,S}''} \sum_{s \in \mathit{states}(M_2)} P_{2,S}'[\mathcal{P}] \frac{P[s] w_2(s, \mathcal{P}')}{P_2'[s]} \mathcal{P}'. \tag{8.74}$$

From (8.68), Expression (8.74) can be rewritten into

$$\sum_{\mathcal{P}' \in \Omega_{3,S}''} \sum_{s \in \mathit{states}(M_2)} \frac{P_2'[s] w_2(s, \mathcal{P}')}{P_2'[s]} \mathcal{P}'. \tag{8.75}$$

After simplifying $P_2'[s]$, since w_2 is a weight function from $\mathcal{P}_2'$ to $\mathcal{P}_{3,S}''$, Expression (8.75) can be rewritten into

$$\sum_{\mathcal{P}' \in \Omega_{3,S}''} P_{3,S}''[\mathcal{P}']\mathcal{P}', \tag{8.76}$$

which can be rewritten into $\mathcal{P}_3''$ using Equation (8.69).

3. For each pair $(s_1', \mathcal{P})$ such that $s_1' \mathrel{\mathcal{R}_1} \mathcal{P}$, $s_1' \mathrel{\mathcal{R}_3} \mathcal{P}_3^{\mathcal{P}}$.

This follows directly from 1 and (8.72).

Let $\mathcal{P}_3'$ be $\mathcal{P}_3''$, and define a new weight function $w : \mathit{states}(M_1) \times \mathit{Probs}(\mathit{states}(M_3)) \to [0, 1]$ such that, for each probability space $\mathcal{P}$ of $\Omega_{2,S}'$, $w(s_1, \mathcal{P}_3^{\mathcal{P}}) = w_1(s_1, \mathcal{P})$. Then, it is easy to check that $\mathcal{P}_1' \sqsubseteq_{\mathcal{R}} \mathcal{P}_{3,S}'$ via w. This fact, together with 2, is sufficient to complete the proof. ■

8.7 Probabilistic Forward Simulations and Trace Distributions

In this section we show that probabilistic forward simulations are sound for the trace distribution precongruence. Specifically, we show that $M_1 \sqsubseteq_{FS} M_2$ implies $M_1 \sqsubseteq_D M_2$. Thus, since $\sqsubseteq_{FS}$ is a precongruence that is contained in $\sqsubseteq_D$, from the definition of $\sqsubseteq_{DC}$ we obtain that $M_1 \sqsubseteq_{FS} M_2$ implies $M_1 \sqsubseteq_{DC} M_2$.

Proposition 8.7.1 *Let $M_1 \sqsubseteq_{FS} M_2$. Then $M_1 \sqsubseteq_D M_2$.*

Proof. Let $\mathcal{R}$ be a probabilistic forward simulation from M_1 to M_2, and let H_1 be a probabilistic execution of M_1 that leads to a trace distribution $\mathcal{D}_1$. From Lemma 8.6.1, there exists a probabilistic execution H_2 of M_2 and two mappings m, S such that (H_1, H_2, m, S) is an execution correspondence structure for $\mathcal{R}$. We show that H_2 leads to a trace distribution $\mathcal{D}_2$ that is equivalent to $\mathcal{D}_1$.

Consider a cone C_β of $\mathcal{D}_1$. The measure of C_β is given by

$$\sum_{q_1 \in states(H_1) | trace(q_1) = \beta, lact(q_1) = lact(\beta)} P_{H_1}[C_{q_1}]. \tag{8.77}$$

The same value can be expressed as

$$\lim_{i \to \infty} \sum_{q_1 \in fringe(H_1, i) | \beta \leq trace(q_1)} P_{H_1}[C_{q_1}]. \tag{8.78}$$

Consider a cone C_β of $\mathcal{D}_2$. The measure of C_β is given by

$$\sum_{q_2 \in states(H_2) | trace(q_2) = \beta, lact(q_2) = lact(\beta)} P_{H_2}[C_{q_2}]. \tag{8.79}$$

The same value can be expressed as

$$\lim_{i \to \infty} \sum_{q_2 \in m(i) | \beta \leq trace(q_2)} P_{m(i)}[C_{q_2}]. \tag{8.80}$$

The reason for the alternative expression is that at the limit each cone of Expression (8.79) is captured completely. Thus, it is sufficient to show that for each finite β and each i,

$$\sum_{q_1 \in fringe(H_1, i) | \beta \leq trace(q_1)} P_{H_1}[C_{q_1}] = \sum_{q_2 \in m(i) | \beta \leq trace(q_2)} P_{m(i)}[q_2]. \tag{8.81}$$

This is shown as follows. Let w_i be the weight function for $m(i) \sqsubseteq_{\mathcal{R}} S(i)$. Then,

$$\sum_{q \in fringe(H_1, i) | \beta \leq trace(q)} P_{H_1}[C_q] = \sum_{q_1 \in fringe(H_1, i) | \beta \leq trace(q_1)} \sum_{\mathcal{P}_2 \in S(i)} w_i(q_1, \mathcal{P}_2). \tag{8.82}$$

Observe that each probability space of $S(i)$ has objects with the same trace, that each state q of *fringe*(H_1, i) is related to some space of $S(i)$, and that each space of $S(i)$ is related to some state q of *fringe*(H_1, i). Thus, from (8.82),

$$\sum_{q \in fringe(H_1, i) | \beta \leq trace(q)} P_{H_1}[C_q] = \sum_{\mathcal{P}_2 \in S(i) | \exists_{q_2 \in \Omega_2} \beta \leq trace(q_2)} \sum_{q_1 \in fringe(H_1, i)} w_i(q_1, \mathcal{P}_2). \tag{8.83}$$

Since w_i is a weight function, we obtain

$$\sum_{q \in fringe(H_1,i) | \beta \leq trace(q)} P_{H_1}[C_q] = \sum_{\mathcal{P}_2 \in S(i) | \exists_{q_2 \in \Omega_2} \beta \leq trace(q_2)} P_{S(i)}[\mathcal{P}_2]. \tag{8.84}$$

Since in a probability space the probability of the whole sample space is 1, we obtain

$$\sum_{q \in fringe(H_1,i) | \beta \leq trace(q)} P_{H_1}[C_q] = \sum_{\mathcal{P}_2 \in S(i) | \exists_{q_2 \in \Omega_2} \beta \leq trace(q_2)} \sum_{q_2 \in \Omega_2} P_{S(i)}[\mathcal{P}_2] P_2[q_2]. \tag{8.85}$$

From an algebraic manipulation based on Condition 3 of an Execution Correspondence Structure, we obtain

$$\sum_{q \in fringe(H_1,i) | \beta \leq trace(q)} P_{H_1}[C_q] = \sum_{q_2 \in m(i) | \beta \leq trace(q_2)} \sum_{\mathcal{P}_2 \in S(i) | q_2 \in \Omega_2} P_{S(i)}[\mathcal{P}_2] P_2[q_2]. \tag{8.86}$$

Finally, from Condition 3 of an Execution Correspondence Structure again, we obtain Equation (8.81). ■

8.8 Discussion

Strong bisimulation was first defined by Larsen and Skou [LS89, LS91] for reactive processes. Successively it was adapted to the alternating model by Hansson [Han94]. In this thesis we have defined the same strong bisimulation as in [Han94]. The formal definition differs from the definition given by Hansson in that we have used the lifting of a relation to probability spaces as defined by Jonsson and Larsen [JL91].

Strong simulation is similar in style to the satisfaction relation for the probabilistic specification systems of Jonsson and Larsen [JL91]. It is from [JL91] that we have borrowed the idea of the lifting of a relation to a probability space.

The probabilistic versions of our simulation relations are justified both by the fact that a scheduler can combine transitions probabilistically, as we have said in this thesis, and by the fact that several properties, namely the ones specified by the logic PCTL of Hansson and Jonsson [Han94], are valid relative to randomized schedulers iff they are valid relative to deterministic schedulers. This fact was first observed by Segala and Lynch [SL94] and can be proved easily using the results about deterministic and randomized schedulers that we proved in Chapter 5.

The weak probabilistic relations were introduced first by Segala and Lynch [SL94]. No simulation relations abstracting from internal computation were defined before. Probabilistic forward simulations are novel in their definition since it is the first time that a state is related to a probability distribution over states.