Uniform Random Process Model Revisited

Wenbo Zhang[1(✉)], Huan Long[1(✉)], and Xian Xu[2(✉)]

[1] BASICS, Shanghai Jiao Tong University, Shanghai, China
{wbzhang,longhuan}@sjtu.edu.cn
[2] East China University of Science and Technology, Shanghai, China
xuxian@ecust.edu.cn

Abstract. Recently, a proper bisimulation equivalence relation for random process model has been defined in a model independent approach. Model independence clarifies the difference between nondeterministic and probabilistic actions in concurrency and makes the new equivalence relation to be congruent. In this paper, we focus on the finite state randomized CCS model and deepen the previous work in two aspects. First, we show that the equivalence relation can be decided in polynomial time. Second, we give a sound and complete axiomatization system for this model. The algorithm and axiomatization system also have the merit of model independency as they can be easily generalized to the randomized extension of any finite state concurrent model.

1 Introduction

Probabilistic processes have been studied for many years as an important extension of classical concurrency theory. Representative work includes the probabilistic extensions of CCS [9,14], the probabilistic CSP [20], the probabilistic ACP [1], and the probabilistic asynchronous π calculus [15].

As being summarized in [8], there are mainly two kinds of channel randomness used in these works. One is generative models [9,17] which bind probabilistic choice to external actions, and the other is reactive models [5,11,19] which interleave nondeterministic choice with probabilistic distributions (i.e., probabilistic choice). The former setup could lead to difficulties in the interleaving of process operations such as composition and restriction. The latter one, however, forces an alternation between nondeterministic choice and probabilistic distribution which brings unnecessary complexity to the system. A different approach is proposed to tackle these problems, by taking a fundamental separation between nondeterministic interaction and probabilistic choice [8]. More specifically, the only probabilistic choice (or random choice) allowed in this new setup is defined as

$$\bigoplus_{i\in I} p_i\tau.T_i \tag{1}$$

where the size of the index set I is at least 2 and $\sum_{i\in I} p_i = 1$. For any (non-probabilistic) process model $\mathbb{M}$, τ is an abstraction for its internal actions. As

A. W. Lin (Ed.): APLAS 2019, LNCS 11893, pp. 388–404, 2019.
https://doi.org/10.1007/978-3-030-34175-6_20

probabilistic choice only happens via τ, (1) ensures the probabilistic choice to be independent of the settings of the original model $\mathbb{M}$. Thus we can uniformly extend $\mathbb{M}$ into its randomized version. In addition to the syntax conciseness, using this extension the probabilistic model will have some elegant algebraic properties. In [8], the author has proposed two examples (processes A, C) to show that (1) helps to overcome the possible confusing caused by traditional syntax, especially under process combinators such as summation, composition, and restriction. At the same time, the corresponding branching bisimilarity relation is shown to be congruent.

Apart from the nice properties brought by this model independent approach, there are still a few issues which require re-investigation, such as equivalence checking and axiomatization. Equivalence checking is one of the important problems in the area of automatic verification. Given two processes E and F of a model, equivalence checking decides whether E and F can be related by a specific equivalence relation. There has been a lot of work on equivalence checking since 1980s [18]. At the same time, axiomatization aims at understanding a language through a set of axioms and inference rules that help to reason about the properties of programs [5]. It is worthwhile to work out a complete axiomatization system for the branching bisimilarity defined in [8].

In this paper we focus on these two problems for randomized model. As a case study, we consider the randomized CCS (Milner's Calculus of Communicating Systems [21]) model. As CCS model is Turing complete the general equivalence checking problem is undecidable, it is standard to consider the finite state submodel [10,22]. Studies on these problems can shed light on the study of other probabilistic process models, such as probabilistic π et al.

The rest of the paper is structured as follows. Section 2 gives preliminary definitions, notational conventions, the random process model and the equivalence congruence; Sect. 3 gives the polynomial equivalence checking algorithm; Sect. 4 axiomatizes the relation of Sect. 2 and shows the soundness and completeness of the axiomatic system; Sect. 5 contains some concluding remarks.

2 Preliminary

Let $Chan$ be the set of channels, ranged over by lowercase letters. The set of nondeterministic actions is denoted as $Act_d = Chan \cup \{\tau\}$, ranged over by small Greek letters. The set of probabilistic actions is $Act_p = \{q\tau \mid 0 < q < 1\}$. $Act = Act_d \cup Act_p$. For a natural number $k \in \mathbb{N}$, we use $[k]$ to denote the set $\{1, 2, \ldots, k\}$.

2.1 Finite State Random Process Model

It is well known that Milner's CCS [21] is Turing complete, which means that RCCS (Randomized CCS) in [8] is also Turing complete as it is an extension of CCS. In order to get any meaningful algorithmic results, as well as what is more

suitable for modeling the reality, we will concentrate on the finite state fragment of the full model, denoted as RCCS_{fs}. The grammar of RCCS_{fs}, is as follows:

$$T := X \mid \sum_{i\in I} \alpha_i.T_i \mid \mu X.T \mid \bigoplus_{i\in I} p_i\tau.T_i \tag{2}$$

In (2), X is a variable. $\sum_{i\in I} \alpha_i.T_i$ means nondeterministic choice term. $\bigoplus_{i\in I} p_i\tau.T_i$ means probabilistic choice term. $\mu X.T$ means fixpoint term. The indexing set I is finite and $\sum_{i\in I} p_i = 1$. We write $\mathbf{0}$ for the nondeterministic term $\sum_{i\in\emptyset} \alpha_i.T_i$ in which $\emptyset$ is the empty set. A trailing $\mathbf{0}$ is often omitted. Particularly, sometimes instead of the standard form $T = \sum_{i\in I} \alpha_i.T_i$ we use $T = T' + \alpha.T''$ to specify one of the summand terms $\alpha.T''$.

A process variable X that appears in $\sum_{i\in I} \alpha_i.T_i$ or $\bigoplus_{i\in I} p_i\tau.T_i$ is guarded, X appears in $a.T_i$ for some visible action a is strongly guarded. We use $\mathrm{fv}(T)$ to stand for the set of variables occurring free (i.e., not bound by μ) in T. A term is a *process* if it contains no free variables. We will use X, Y, Z for process variables and $A, B, C, D, E, F, G, H, L$ for processes. The set of all RCCS_{fs} processes (terms resp.) will be represented by $\mathcal{P}_{\mathrm{RCCS}_{fs}}$ ($\mathcal{T}_{\mathrm{RCCS}_{fs}}$ resp.). Comparing to the definition in [8], we drop the composition operation for its combination with fixpoint operator could lead to processes with infinite state. A simple counterexample is $\mu X.(s + t|\tau.X)$.

The transition semantics of RCCS_{fs} is generated by the following labelled transition rules, where $\lambda \in Act$:

$$\frac{}{X \xrightarrow{X} \mathbf{0}} \qquad \frac{}{\sum_{i\in I} \alpha_i.T_i \xrightarrow{\alpha_i} T_i} \qquad \frac{}{\bigoplus_{i\in I} p_i\tau.T_i \xrightarrow{p_i\tau} T_i}$$

$$\frac{T\{\mu X.T/X\} \xrightarrow{\lambda} T'}{\mu X.T \xrightarrow{\lambda} T'} \tag{3}$$

Follow the convention used in [8], for an equivalence relation $\mathcal{E}$ on $\mathcal{P}_{\mathrm{RCCS}_{fs}}$, we write $A\mathcal{E}B$ for $(A, B) \in \mathcal{E}$. The notation $\mathcal{P}_{\mathrm{RCCS}_{fs}}/\mathcal{E}$ stands for the set of equivalence classes defined by $\mathcal{E}$. The equivalence class containing A is denoted by $[A]_{\mathcal{E}}$. For $\mathcal{C} \in \mathcal{P}_{\mathrm{RCCS}_{fs}}/\mathcal{E}$ we write $A \xrightarrow{l} \mathcal{C}$ for the fact that $A \xrightarrow{l} A' \in \mathcal{C}$ for some A'.

We use $\mathcal{T}_v$ to stand for terms that are actually a variable (the one in the first rule). Terms that can immediately do a nondeterministic choice (as in the second rule) are called *nondeterministic terms*, denoted as $\mathcal{T}_d$. Terms that can immediately do a probabilistic action are called *probabilistic terms*, denoted as $\mathcal{T}_p$ (as in the last rule). It is obvious that $\mathcal{T}_{\mathrm{RCCS}_{fs}} = \mathcal{T}_d \cup \mathcal{T}_p \cup \mathcal{T}_v$.

For terms S and T, if S can be transformed into T via one or a sequence of rules in (3), we say that S can *reach* T, or equivalently, T is *reachable* from S.

Given $S \in \mathcal{P}_{\mathrm{RCCS}_{fs}}$, we use R_S to stand for the set of process expressions reachable from S. The following proposition justifies the finite state property of the model defined in (2).

Proposition 1. *Given $S \in \mathcal{P}_{RCCS_{fs}}$, R_S is finite.*

The proposition can be proved by induction on the grammar depth which is standard. We omit the details here.

2.2 Branching Bisimulation Congruence

Here we give the bisimulation relation for which we will study the equivalence checking algorithm and axiomatization. For self-containment, we include relating definitions in this section.

The collective silent transition is firstly introduce in [8]:

$$\bigoplus_{i\in I} p_i\tau.T_i \xrightarrow{\coprod_{i\in I} p_i\tau} \coprod_{i\in I} T_i$$

Definition 1 (ϵ-tree [8]). *Let $A \in \mathcal{P}_{RCCS_{fs}}$ be a process and $\mathcal{E}$ be an equivalence relation on $\mathcal{P}_{RCCS_{fs}}$ An* ϵ-tree $t^A_{\mathcal{E}}$ of A with regard to $\mathcal{E}$ *is a labeled-tree such that the following statements hold true.*

- *Every node of $t^A_{\mathcal{E}}$ is labeled by elements of $[A]_{\mathcal{E}}$. The root is labeled by A.*
- *The edges are labeled by elements of $(0, 1]$.*
- *If an edge from a node B to a node B' is labeled p for some $p \in (0, 1)$, then some collective silent transition $B \xrightarrow{\coprod_{i\in[k]} p_i\tau} \coprod_{i\in[k]} B_i$ exists such that for every $i \in [k]$, there exists an edge from B to B_i labeled p_i, and $B_1, \ldots, B_k$ are the only children of B.*
- *If an edge from a node B to a node B' is labeled 1, then $B \xrightarrow{\tau} B'$ and B' is the only child of B.*

Intuitively epsilon-tree is a random version of a sequence of state-preserving internal actions. Sometimes we will use t instead of $t^A_{\mathcal{E}}$ for simplicity when A and $\mathcal{E}$ are unstressed in the context.

A *branch* in an ϵ-tree t is either a finite path going from the root to a leaf or an infinite path. The length $|\pi|$ of a branch π is the number of edges in π if π is finite; it is ω otherwise. For $i \leq |\pi|$ let $\pi(i)$ be the label of the i-th edge. The probability $\mathbb{P}(\pi)$ of a finite branch π is $\prod_{i\leq|\pi|} \pi(i)$. A branch of length zero is a single node, and its probability is 1. The probability of an infinite path $A \xrightarrow{p_1} \xrightarrow{p_2} \ldots \xrightarrow{p_k} \ldots$ is $\lim_{k\to\infty} \prod_{i<k} p_i$.

Given an ϵ-tree t, the probability of the finite branches of t is defined by $\mathbb{P}^f(t) = \lim_{k\to\infty} \mathbb{P}^k(t)$, where

$$\mathbb{P}^k(t) = \sum\{\mathbb{P}(\pi) \mid \pi \text{ is a finite branch in } t \text{ such that } |\pi| \leq k\}.$$

An ϵ-tree $t^A_{\mathcal{E}}$ is *regular* if $\mathbb{P}^f(t^A_{\mathcal{E}}) = 1$.

Definition 2 (l-transition [8]). *For $l \in Act_d$ and $\mathcal{B} \in \mathcal{P}_{RCCS_{fs}}/\mathcal{E}$, suppose $l \neq \tau \vee \mathcal{B} \neq [A]_{\mathcal{E}}$. An* l-transition *from A to $\mathcal{B}$ with regard to $\mathcal{E}$ consists of a regular ϵ-tree $t^A_{\mathcal{E}}$ of A with regard to $\mathcal{E}$ and a transition $L \xrightarrow{l} L' \in \mathcal{B}$ for every leaf L of $t^A_{\mathcal{E}}$. We will write $A \rightsquigarrow_{\mathcal{E}} \xrightarrow{l} \mathcal{B}$ if there is an l-transition from A to $\mathcal{B}$ with regard to $\mathcal{E}$.*

Intuitively l-transition characterizes that after some state-preserving silent transitions, an l-action is performed and the resulting processes should be in the same equivalence class.

Suppose $L \xrightarrow{\coprod_{i\in[k]} p_i\tau} \coprod_{i\in[k]} L_i$ such that $\exists i \in [k]$, $L_i \in \mathcal{B} \neq [L]_\mathcal{E}$. We define

$$\mathbb{P}(L \xrightarrow{\coprod_{i\in[k]} p_i\tau} \mathcal{B}) = \sum\{p_i | L \xrightarrow{p_i\tau} L_i \in \mathcal{B} \wedge i \in [k]\}$$

Define the weighted probability

$$\mathbb{P}_\mathcal{E}(L \xrightarrow{\coprod_{i\in[k]} p_i\tau} \mathcal{B}) = \mathbb{P}(L \xrightarrow{\coprod_{i\in[k]} p_i\tau} \mathcal{B})/(1 - \mathbb{P}(L \xrightarrow{\coprod_{i\in[k]} p_i\tau} [L]_\mathcal{E}))$$

Definition 3 (q-transition [8]). *A q-transition from A to $\mathcal{B}$ with regard to $\mathcal{E}$ consists of a regular ϵ-tree $t^A_\mathcal{E}$ of A with regard to $\mathcal{E}$ and for every leaf L of $t^A_\mathcal{E}$, a collective silent transition $L \xrightarrow{\coprod_{i\in[k]} p_i\tau} \coprod_{i\in[k]} L_i$ such that $\mathbb{P}_\mathcal{E}(L \xrightarrow{\coprod_{i\in[k]} p_i\tau} \mathcal{B}) = q$. We use $A \rightsquigarrow_\mathcal{E} \xrightarrow{q} \mathcal{B}$ to mean there is a q-transition from A to $\mathcal{B}$ with regard to the relation $\mathcal{E}$.*

Intuitively q-transition characterizes that after some state-preserving silent transitions, random choices with total conditional probability q are performed and the resulting processes should be in the same equivalence class.

Definition 4. *[8] An equivalence $\mathcal{E}$ on $\mathcal{P}$ is a* branching bisimulation *if* $(1, 2)$ *are valid.*

1. *If $B\mathcal{E}A \rightsquigarrow_\mathcal{E} \xrightarrow{l} \mathcal{C} \in \mathcal{P}/\mathcal{E}$ such that $l \neq \tau \vee \mathcal{C} \neq [A]_\mathcal{E}$, then $B \rightsquigarrow_\mathcal{E} \xrightarrow{l} \mathcal{C}$.*
2. *If $B\mathcal{E}A \rightsquigarrow_\mathcal{E} \xrightarrow{q} \mathcal{C} \in \mathcal{P}/\mathcal{E}$ such that $\mathcal{C} \neq [A]_\mathcal{E}$, then $B \rightsquigarrow_\mathcal{E} \xrightarrow{q} \mathcal{C}$.*

Finally we can define the equality on $\mathcal{P}_{\mathrm{RCCS}_{fs}}$. It is the largest branching bisimulation on $\mathcal{P}_{\mathrm{RCCS}_{fs}}$, denoted by $\simeq_{\mathrm{RCCS}_{fs}}$. Sometimes we will use $\simeq$ instead of $\simeq_{\mathrm{RCCS}_{fs}}$ for simplicity when its meaning is clear from the context.

The following proposition is a special case of the Theorem 17 in [8]. Here we present it without proof.

Proposition 2. *The equality $\simeq_{RCCS_{fs}}$ is a congruence.*

3 Equivalence Checking Algorithm

Equivalence checking is one of the key problems in verification. It gives the answer whether two systems are related by a given equivalent relation. As far as branching bisimulation is concerned, some representative includes [7,16]. Meanwhile, to the probabilistic process calculus model, there are also some interesting work such as [3,23].

Here we develop an algorithm to decide the equivalence relation $\simeq_{\mathrm{RCCS}_{fs}}$ for RCCS_{fs} processes. Recall that for a given random process A, we use R_A to denote the set of all processes reachable from A. In Proposition 1, we have

already known that this set is finite, here we will further show that it can be constructed in polynomial time with respect to the length of the given process. Pseudocode of our algorithm is given in Algorithm 1.

Algorithm 1. Compute R_A

Input: $A \in \mathcal{P}_{\mathrm{RCCS}_{fs}}$
Output: R_A
1: $R_A \coloneqq \emptyset$, $R' \coloneqq \{A\}$
2: **while** $R' \neq \emptyset$ **do**
3: Choose a process B from R', $R' \coloneqq R' - \{B\}$, $R_A \coloneqq R_A \cup \{B\}$
4: **if** $B = \sum_{i\in I} \alpha_i.T_i$ **then**
5: $R' \coloneqq R' \cup \{T_i : i \in I\} \setminus R_A$
6: **else if** $B = \bigoplus_{i\in I} p_i\tau.T_i$ **then**
7: $R' \coloneqq R' \cup \{T_i : i \in I\} \setminus R_A$
8: **else if** $B = \mu X.T$ **then**
9: $R' \coloneqq R' \cup \{T\{\mu X.T/X\}\} \setminus R_A$
10: **return** R_A

For a better understanding of the algorithm, we give one simple example here.

Example 1. Let $H = \mu X.(\frac{1}{2}\tau.(a + \tau.X) \oplus \frac{1}{2}\tau.(b + \tau.X))$, then Algorithm 1 will return

$$R_H = \left\{H, \frac{1}{2}\tau.(a + \tau.H) \oplus \frac{1}{2}\tau.(b + \tau.H), a + \tau.H, b + \tau.H, \mathbf{0}\right\}.$$

As usual, a *partition* of process set $\mathcal{P}$ is a collection of $\mathcal{X}$ containing pairwise disjoint subsets of $\mathcal{P}$ such that each element $A \in \mathcal{P}$ is contained in some $\mathcal{C} \in \mathcal{X}$. The equivalence class containing A is denoted by $[A]_{\mathcal{X}}$. Let $\mathcal{E}_{\mathcal{X}}$ be the equivalence relation induced by the partition $\mathcal{X}$. Given two partitions $\mathcal{X}_1$ and $\mathcal{X}_2$ of the same set. We say $\mathcal{X}_1$ is coarser than $\mathcal{X}_2$ (or equivalently $\mathcal{X}_2$ is finer than $\mathcal{X}_1$) if every element in $\mathcal{X}_2$ is a subset of some element in $\mathcal{X}_1$.

Next we propose a technical definition which is closely related to the conception of ϵ-tree given in Definition 1.

Definition 5. *The* ϵ-graph of A with regard to an equivalence relation $\mathcal{E}$ *is a weighted directed graph, denoted by* $G^A_{\mathcal{E}}$. $G^A_{\mathcal{E}}$ *is defined by merging nodes of the same name from an* ϵ*-tree* $t^A_{\mathcal{E}}$ *into one node. A vertex in* $G^A_{\mathcal{E}}$ *is called a* sink *node if its out degree is 0. Let* $sn(G^A_{\mathcal{E}})$ *be the set of all sink nodes of* $G^A_{\mathcal{E}}$.

For given process A and $\mathcal{E}$, though there could be infinitely many different ϵ-trees, the number of all possible ϵ-graphs of A with regard to $\mathcal{E}$ is finite.

Proposition 3. *Let* $\mathcal{P}$ *be a process set and* $\mathcal{E}$ *be an equivalence relation. For a process* $A \in \mathcal{P}$*, and a process set* $\mathcal{P}' \subseteq \mathcal{P}$*, there exists a regular* ϵ*-tree* $t^A_{\mathcal{E}}$ *with leaf nodes set* $\mathcal{P}'$ *(all of the same name) if and only if there exists an* ϵ*-graph* $G^A_{\mathcal{E}}$ *with a sink node named* $\mathcal{P}'$.

By Proposition 3, the transformation from ϵ-tree to ϵ-graph will not affect the bisimilarity relation. Yet ϵ-graph is technically more convenient for presenting our equivalence checking algorithm.

We put an example here to explain the difference between ϵ-tree and ϵ-graph.

Example 2. For the process H in Example 1, a branching bisimulation for $\mathcal{P}_H$ is the equivalence $\mathcal{E}$ rendering the truth that $[H]_{\mathcal{E}} = [a + \tau.H]_{\mathcal{E}} = [b + \tau.H]_{\mathcal{E}}$. For the ϵ-tree in Fig. 1(a), the corresponding ϵ-graph has a sink node $b + \tau.H$. For the second one in Fig. 1(b) the sink node is $a + \tau.H$. For the ϵ-tree in Fig. 1(c), there does not exist a visible action that all leaves can immediately do, and there does not exist a sink node for the ϵ-graph.

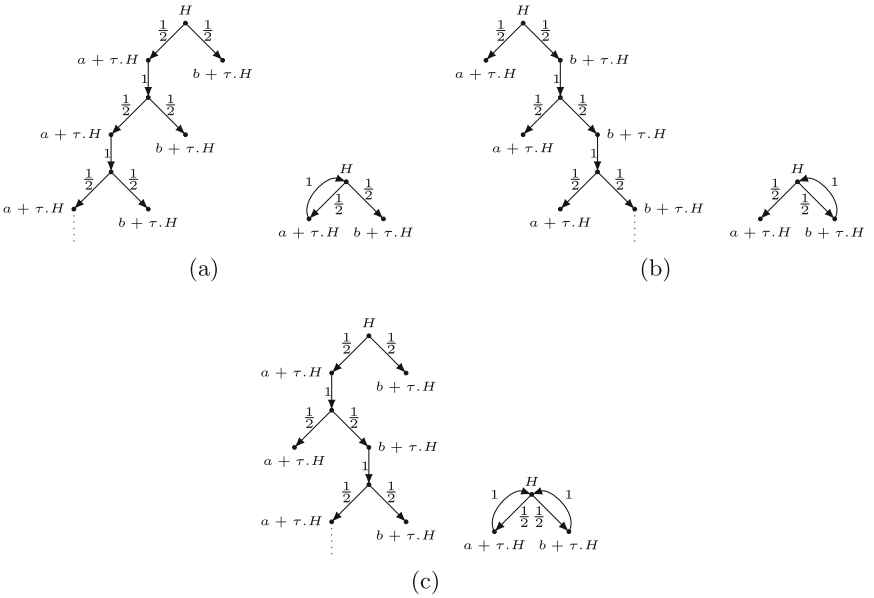

Fig. 1. ϵ-trees and corresponding ϵ-graphs

Here we will introduce one more convention for the description of our algorithm. We use the symbol $\hat{\varphi}\tau$ to represent *any* $p\tau$ where $p \in (0, 1]$. In other words, using $\hat{\varphi}\tau$ means we are talking about a probabilistic action without specifying the concrete probability value.

Definition 6. *Let $\mathcal{X}$ be a partition of process set $\mathcal{P}$. A* splitter *of a partition $\mathcal{X}$ is a triple $(\mathcal{C}_1, l, \mathcal{C}_2)$ consisting of $\mathcal{C}_1, \mathcal{C}_2 \in \mathcal{X}$ and an action $l \in Act_d \cup \{\hat{\varphi}\tau\}$. One of the following statements is valid:*

1. *If $l \in Act_d$, and $\mathcal{C}_1 \neq \mathcal{C}_2$ when $l = \tau$, then there exist some $A, A' \in \mathcal{C}_1$, such that for exactly one of A, A', there is an ϵ-graph $G^A_{\mathcal{X}}$ ($G^{A'}_{\mathcal{X}}$ resp.), all of $sn(G^A_{\mathcal{X}})$ ($sn(G^{A'}_{\mathcal{X}})$ resp.) can do an immediate l action to $\mathcal{C}_2$.*

2. *If* $l = \hat{\varphi}\tau$, *and* $\mathcal{C}_1 \neq \mathcal{C}_2$, *then there exist* $A, A' \in \mathcal{C}_1$ *and* $q \in (0,1]$, *such that for exactly one of* A, A', *there is an* ϵ*-graph* $G^A_{\mathcal{X}}$ ($G^{A'}_{\mathcal{X}}$ *resp.), for any* $T \in sn(G^A_{\mathcal{X}})$ ($sn(G^{A'}_{\mathcal{X}})$ *resp.),* $\mathbb{P}_{\mathcal{E}_{\mathcal{X}}}(T \xrightarrow{\amalg_{i\in[k]}\, p_i\tau} \mathcal{C}_2) = q$.

Intuitively speaking, our equivalence checking strategy starts with a finite set which contains all reachable states for a pair of processes (the coarsest partition). Keep refining the current sets into finer ones according to their one-step difference until no further refinement is possible (the finest partition), where states of the same set are equivalent to each other.

In detail, to refine the partition $\mathcal{X}$, according to Definition 6, there are two cases to be considered:

1. If $\mathcal{X}$ has a splitter $(\mathcal{C}_1, \hat{\varphi}\tau, \mathcal{C}_2)$.
   Let $tn(\mathcal{P}, \mathcal{X}) \subseteq \mathcal{P}$ be the set composed of all processes that can perform probabilistic τ step into a different class with nonzero probability. Firstly, we split $\mathcal{C}_1 \cap tn(\mathcal{P}, \mathcal{X})$ into $\mathcal{C}_1 \cap tn(\mathcal{P}, \mathcal{X})/ =_p$, where $A =_p A'$ iff $\mathbb{P}_{\mathcal{X}}(A \xrightarrow{\amalg_{i\in[k]}\, p_i\tau} \mathcal{C}_2) = \mathbb{P}_{\mathcal{X}}(A' \xrightarrow{\amalg_{i\in[k]}\, p_i\tau} \mathcal{C}_2)$. Then, an equivalent class $\mathcal{B} \in \mathcal{C}_1 \cap tn(\mathcal{P}, \mathcal{X})/ =_p$ is enriched with process $B \in \mathcal{C}_1 \setminus tn(\mathcal{P}, \mathcal{X})$ which satisfies (denoted as Δ) :
   (a) There exists an ϵ-graph $G^B_{\mathcal{X}}$ with $sn(G^B_{\mathcal{X}}) \subseteq \mathcal{B}$.
   (b) For any other $\mathcal{B}' \in \mathcal{C}_1 \cap tn(\mathcal{P}, \mathcal{X})/ =_p$, there does not exist an ϵ-graph $G^B_{\mathcal{X}}$ with $sn(G^B_{\mathcal{X}}) \subseteq \mathcal{B}'$.
   Let $\overline{\mathcal{B}} \stackrel{\text{def}}{=} \mathcal{B} \cup \{B : B \in \mathcal{C}_1, B \text{ satisfies } \Delta\}$ be the closure of $\mathcal{B}$. We put the remaining processes into
   $Res(\mathcal{C}_1) \stackrel{\text{def}}{=} \{C \in \mathcal{C}_1 : C \text{ does not satisfies } \Delta \text{ for any } \mathcal{B} \in \mathcal{C}_1 \cap tn(\mathcal{P}, \mathcal{X})/ =_p\}$.
   Formally, the strategy we used for refining $\mathcal{X}$ via a splitter $(\mathcal{C}_1, p\tau, \mathcal{C}_2)$ is:

$$\begin{aligned}\mathbf{Refine}(\mathcal{X}, (\mathcal{C}_1, \tau, \mathcal{C}_2)) \stackrel{\text{def}}{=} &(\mathcal{X} \setminus \{\mathcal{C}_1\}) \cup \{\overline{\mathcal{B}} : \mathcal{B} \in \mathcal{C}_1 \cap tn(\mathcal{P}, \mathcal{X})/ =_p\} \\ &\cup (\{Res(\mathcal{C}_1)\} \setminus \{\emptyset\}).\end{aligned}$$

2. If $\mathcal{X}$ has a splitter $(\mathcal{C}_1, \alpha, \mathcal{C}_2)$, $\alpha \in Act_d$, and $\mathcal{C}_1 \neq \mathcal{C}_2$ when $\alpha = \tau$.
   Let $\mathcal{D} \stackrel{\text{def}}{=} \{B \in \mathcal{C}_1 :$ there exists an ϵ-graph $G^B_{\mathcal{X}}$, all of $sn(G^B_{\mathcal{X}})$ can do an immediate α action to $\mathcal{C}_2\}$. We can define the method for refining $\mathcal{X}$ via a splitter $(\mathcal{C}_1, \alpha, \mathcal{C}_2)$:

   $\mathbf{Refine}(\mathcal{X}, (\mathcal{C}_1, \alpha, \mathcal{C}_2)) \stackrel{\text{def}}{=} (\mathcal{X} \setminus \{\mathcal{C}_1\}) \cup \mathcal{D} \cup (\mathcal{C}_1 \setminus \mathcal{D})$.

Note that for every partition $\mathcal{X}$ which is coarser than $\mathcal{P}/\simeq$ and every nonempty splitter $(\mathcal{C}_1, l, \mathcal{C}_2)$ of $\mathcal{X}$, the partition $\mathbf{Refine}(\mathcal{X}, (\mathcal{C}_1, l, \mathcal{C}_2))$ is no finer than $\mathcal{P}/\simeq$ while strictly finer than $\mathcal{X}$. If there is no splitter for $\mathcal{X}$ (i.e., if neither of the above two cases applies), then through proof by contradiction, it can be easily concluded that $\mathcal{X} = \mathcal{P}/\simeq$. This analysis turns out to be the proof of the following proposition.

Proposition 4. *Let* $\mathcal{X}$ *be a partition of process set* $\mathcal{P}$. *If* $\mathcal{X}$ *cannot be refined anymore, then* $\mathcal{X} = \mathcal{P}/\simeq$.

This justifies the correctness of Algorithm 2.

Algorithm 2. Equivalence Checking Algorithm

Input: Process A , B
Output: Is $A \simeq B$?
1: Compute $R \coloneqq R_A \cup R_B$
2: $\mathcal{X} \coloneqq \{R\}$
3: **while** $\mathcal{X}$ contains a splitter $(\mathcal{C}_1, l, \mathcal{C}_2)$ **do**
4: $\quad \mathcal{X} \coloneqq$ **Refine**$(\mathcal{X}, (\mathcal{C}_1, l, \mathcal{C}_2))$
5: **if** $[A]_{\mathcal{X}} = [B]_{\mathcal{X}}$ **then**
6: $\quad$ **return** true
7: **else**
8: $\quad$ **return** false

Theorem 1. *$\simeq_{RCCS_{fs}}$ can be decided in polynomial time.*

Proof. There exists a constant c, such that the numbers of elements in R is bounded by $c \cdot (|A| + |B|)$ and for process $E \in R$, $|E| < c \cdot (|A| + |B|)$. The **while** loop of line 3–4 can be repeated at most $c \cdot (|A| + |B|)$ times. For each l and $\mathcal{C}_2$, we can construct the process set $S = \{A \mid A \in \mathcal{C}_1, A \xrightarrow{l} \mathcal{C}_2\}$ in $\mathcal{O}((|A| + |B|)^3)$ time and then decide the condition in line 3 by searching for an ϵ-graph with sink nodes in S. It can be done by depth first search in $\mathcal{O}((|A| + |B|)^3)$ time. Overall the algorithm will terminate in $\mathcal{O}((|A| + |B|)^4)$ time.

4 Axiomatizations

4.1 Discussion of the Axioms

In the original CCS model, a complete axiomatization for branching bisimulation congruence of finite process will first convert any expression into a strongly guarded one. If two strongly guarded expressions are branching bisimilar, they can be proved to be equal in axiomatic system [10]. However, in probabilistic model, there exist some expressions that cannot be transformed to a strongly guarded one, e.g., $\mu X(\tau.(\frac{1}{2}\tau.X \oplus \frac{1}{2}\tau.b) + a)$. It means that a τ-loop containing probabilistic τ may be not state-preserving under $\simeq$. We will define *probabilistically guarded*. Intuitively X is probabilistically guarded in T if T can not do some τ actions to X with probability 1.

Definition 7. *The variable X is probabilistically guarded in T if at least one of the following statements is true:*

- *There is no free occurrence of X in T, or every free occurrence of X in T occurs within some subexpression $a.F$.*
- *If $T \in \mathcal{T}_d$, then for any term T' such that $T \xrightarrow{\tau} T'$, X is probabilistically guarded in T'.*

– *If* $T \in \mathcal{T}_p$, *then there exists a term* T' *such that* $T \xrightarrow{p\tau} T'$, X *is probabilistically guarded in* T'.

Otherwise X *is probabilistically unguarded in* T.

Example 3. X is probabilistically guarded in $\frac{1}{2}\tau.(\frac{1}{2}\tau.a \oplus \frac{1}{2}\tau.X) \oplus \frac{1}{2}\tau.(\frac{1}{2}\tau.X \oplus \frac{1}{2}\tau.X)$. X is probabilistically unguarded in $a.X + \tau.X$.

If for every occurrence of $\mu X.T$ in E, X is probabilistically guarded in T, we call process E is probabilistically guarded. Let $\mathcal{P}^g_{\mathrm{RCCS}_{fs}}$ be the set of probabilistically guarded processes.

The axioms that characterize the equivalence relation given in Sect. 2.2 are listed below. We will prove that this set of axioms is sound and complete for the relation $\simeq_{\mathrm{RCCS}_{fs}}$.

$E1$ $T = T$
$E2$ if $S = T$ then $T = S$
$E3$ if $S = T$ and $T = R$ then $S = R$
$E4$ if $S_i = T_i$ for each $i \in I$ then $\sum_{i\in I} \alpha_i.S_i = \sum_{i\in I} \alpha_i.T_i$
$E5$ if $S_i = T_i$ for each $i \in I$ and $\sum_{i\in I} p_i = 1$, then $\bigoplus_{i\in I} p_i\tau.S_i = \bigoplus_{i\in I} p_i\tau_i.T_i$
$E6$ if $S = T$ then $\mu X.S = \mu X.T$

$A1$ $\bigoplus_{i\in I} p_i\tau.S_i \oplus p\tau.S \oplus q\tau.S = \bigoplus_{i\in I} p_i\tau.S_i \oplus (p+q)\tau.S$, $p + q < 1$
$A2$ $p\tau.S \oplus q\tau.S = \tau.S$

$B1$ $\left(\sum_{i\in I'\subseteq I} \alpha_i.S_i\right) + \tau.(\sum_{i\in I} \alpha_i.S_i) = \sum_{i\in I} \alpha_i.S_i$
$B2$ if $\frac{p_1}{q_1} = \cdots = \frac{p_i}{q_i} < 1$ and $\sum_{i\in I} q_i = 1$,
then $\bigoplus_{i\in I} p_i\tau.S_i \oplus p\tau.(\bigoplus_{i\in I} q_i\tau.S_i) = \bigoplus_{i\in I} q_i\tau.S_i$, $p = 1 - \sum_{i\in I} p_i$

$R1$ $\mu X.T = T\{\mu X.T/X\}$
$R2$ if $S = T\{S/X\}$ then $S = \mu X.T$, provided X is probabilistically guarded in T
$R3$ $\mu X.(\tau.X + \sum_{i\in I} \alpha_i.T_i) = \mu X(\sum_{i\in I} \alpha_i.T_i)$
$R4$ $\mu X.(\tau.(\tau.S + \sum_{i\in I} \alpha_i.T_i) + \sum_{j\in J} \beta_j.R_j) = \mu X(\tau.S + \sum_{i\in I} \alpha_i.T_i + \sum_{j\in J} \beta_j.R_j)$,
provided X is probabilistically unguarded in S

One writes $\mathcal{A} \vdash E = F$, with $\mathcal{A}$ a list of axiom names, if the equation $E = F$ is derivable from the axioms in $\mathcal{A}$. In this paper, we take the convention that $E1-6$ and $A1-2$ are always in $\mathcal{A}$.

Comparing to the earlier work on axiomatization for probabilistic bisimulation [2,4,6,13,17,24], $B2$ highlights the nucleus of the model independent approach for random process model. That is, instead of the absolute probability value (or probability distribution), we use the *weighted* probability in [8], which basically characterizes the conditional probability of transferring from one state to another.

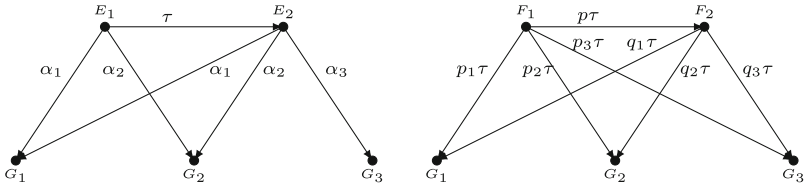

Fig. 2. State-preserving τ and $p\tau$ actions

Axioms $B1$, $B2$ are both motivated by the axiom B in [10]. $B1$ modifies B in two aspects: Firstly, it does not use the heading external action, as the grammar ensures the terms are weak guarded; Secondly, it uses summation of a set of terms rather than binary summation. The intuition of axiom $B1$ is showed in Fig. 2(a) [12]. The τ action $E_1 \xrightarrow{\tau} E_2$ is state-preserving if process E_2 can do any actions E_1 can do. $B2$ is a random extension of $B1$. The intuition of axiom $B2$ is showed in Fig. 2(b). The probabilistic τ action $F_1 \xrightarrow{p\tau} F_2$ is state-preserving if process F_2 can do exactly F_1 can do with the same *weighted* probability($\frac{p_i}{q_i}$ is a constant value for $i \in \{1,2,3\}$).

The presentation of our work on soundness and completeness follows a similar strategy as in [10]. One can refer to van Glabbeek's paper for a comparison.

4.2 Soundness

The soundness of $E1-6$ has been validated by Proposition 2. The soundness of $A1-2$, $B1$ and $R3$ can be easily shown by the definition of the equivalence relation. The soundness of $R1$ follows from the fact that $\mu X.T \xrightarrow{\alpha} F \iff T\{\mu X.T/X\} \xrightarrow{\alpha} F$. The soundness of the remaining axioms are given below.

Proposition 5 (Soundness of $B2$). *If* $\frac{p_1}{q_1} = \cdots = \frac{p_i}{q_i} < 1$ *and* $\sum_{i\in I} q_i = 1$, *then* $\bigoplus_{i\in I} p_i\tau.E_i \oplus p\tau.(\bigoplus_{i\in I} q_i\tau.E_i) \simeq \bigoplus_{i\in I} q_i\tau.E_i$, $p = 1 - \sum_{i\in I} p_i$.

Proof. Let $F_1 = \bigoplus_{i\in I} q_i\tau.E_i$ and $F_2 = \bigoplus_{i\in I} p_i\tau.E_i \oplus p\tau.(\bigoplus_{i\in I} q_i\tau.E_i)$. We consider the following two cases:

- $\forall i \in I$, $E_i \simeq F_1$.
  For every ϵ-tree $t_{\simeq}^{F_j}$, $j \in \{1,2\}$, we can construct an ϵ-tree $t_{\simeq}^{F_{3-j}}$ with the same set of leaf nodes of $t_{\simeq}^{F_j}$. Thus $F_1 \simeq F_2$.
- $\exists i \in I$, $E_i \not\simeq F_1$.
  Let $I' \subseteq I$ be the set of indices satisfying $E_{i'} \not\simeq F_1$, $i' \in I'$. Let $r_{i'} = \frac{q_{i'}}{\sum_{i'\in I'} q_{i'}}$, then $F_j \rightsquigarrow_{\mathcal{E}} \xrightarrow{r_{i'}} [E_{i'}]_{\simeq}$ for $j \in \{1,2\}$.

Definition 8. *A* branching bisimulation up to $\simeq$ *is a symmetric relation* $\mathcal{R} \subseteq \mathcal{P}_{RCCS_{fs}} \times \mathcal{P}_{RCCS_{fs}}$ *such that*

- *if* $E\mathcal{R}F$ *and* $E \rightsquigarrow_{\simeq} \xrightarrow{l} \mathcal{B}_1$ *such that* $l \neq \tau \vee \mathcal{B}_1 \neq [E]_{\simeq}$*, then there exists* E', F' *such that* $E' \in \mathcal{B}_1 \wedge F' \in \mathcal{B}_2 \wedge F \rightsquigarrow_{\simeq} \xrightarrow{l} \mathcal{B}_2 \wedge (E', F') \in \mathcal{R}$*.*
- *if* $E\mathcal{R}F$ *and* $E \rightsquigarrow_{\simeq} \xrightarrow{q} \mathcal{B}_1$ *such that* $\mathcal{B}_1 \neq [E]_{\simeq}$*, then there exists* E', F' *such that* $E' \in \mathcal{B}_1 \wedge F' \in \mathcal{B}_2 \wedge F \rightsquigarrow_{\simeq} \xrightarrow{q} \mathcal{B}_2 \wedge (E', F') \in \mathcal{R}$*.*

Proposition 6. *If* $\mathcal{R}$ *is a branching bisimulation up to* $\simeq$ *and* $E\mathcal{R}F$*, then* $E \simeq F$*.*

Proposition 7. *Variable* X *is probabilistically guarded in a term* $T \in \mathcal{T}_{RCCS_{fs}}$*. If there is an* l*-transition or* q*-transition from* $T\{F/X\}$ *to* $\mathcal{B}$*, then there is a process* $T'\{F/X\} \in \mathcal{B}$ *such that* T' *is reachable from* T*.*

Proof. Induction on the structure of T.

Proposition 8 (Soundness of $R2$**).** *If* $F \simeq S\{F/X\}$*, then* $F \simeq \mu X.S$*, provided* X *is probabilistically guarded in* S*.*

Proof. Consider the following relation

$$\mathcal{R} = \Big\{ \big(T\{F/Y\}, T\{\mu X.S/Y\}\big) \;\Big|\; \mathrm{fv}(T) = \{Y\} \Big\}.$$

Then $(F, \mu X.S) \in \mathcal{R}$. By Proposition 6, it suffices to prove that the symmetric closure of $\mathcal{R}$ is a branching bisimulation up to $\simeq$.

- If $T\{F/Y\} \rightsquigarrow_{\simeq} \xrightarrow{l} \mathcal{B}_1$ such that $l \neq \tau \vee \mathcal{B}_1 \neq [T\{F/Y\}]_{\simeq}$.
  Consider the ϵ-tree $t_{\simeq}^{T\{F/Y\}}$, we can construct an ϵ-tree $t_{\simeq}^{T\{\mu X.S/Y\}}$ from $t_{\simeq}^{T\{F/Y\}}$ by recursively replace the subtree from node F with an ϵ-tree $t_{\simeq}^{S\{F/X\}}$.
  If there is a process $E' = T'\{F/Y\} \in \mathcal{B}_1$ such that T' is reachable from T, then $E'\mathcal{R}F' = T'\{\mu X.S/Y\} \in \mathcal{B}_2$ and $T\{\mu X.S/Y\} \rightsquigarrow_{\simeq} \xrightarrow{l} \mathcal{B}_2$.
  Otherwise, every branch of $t_{\simeq}^{T\{F/Y\}}$ steps into F. Let $E' \in \mathcal{B}_1$, E' is reachable from F. Since $F \simeq S\{F/X\}$, and X is probabilistically guarded in S, by Proposition 7, there is a process $E''\{F/X\} \simeq E'$ where E'' is reachable from S. What's more, there is a process $F' = E''\{\mu X.S/X\} \in \mathcal{B}_2$ and $T\{\mu X.S/Y\} \rightsquigarrow_{\simeq} \xrightarrow{l} \mathcal{B}_2$. Then we have

  $$(E''\{F/X\}, F') = (E''\{Y/X\}\{F/Y\}, E''\{\mu X.Y/X\}\{\mu X.S/Y\}) \in \mathcal{R}$$

- If $T\{F/Y\} \rightsquigarrow_{\simeq} \xrightarrow{q} \mathcal{B}_1$ such that $\mathcal{B}_1 \neq [T\{F/Y\}]_{\simeq}$.
  Similar with the case of l-transition.

Proposition 9 (Soundness of $R4$). $\mu X.(\tau.(\tau.S + \sum_{i\in I}\alpha_i.T_i) + \sum_{j\in J}\beta_j.R_j) \simeq \mu X(\tau.S + \sum_{i\in I}\alpha_i.T_i + \sum_{j\in J}\beta_j.R_j)$, *provided* X *is probabilistically unguarded in term* E.

Proof. Let

$$A_1 = \mu X.(\tau.(\tau.S + \sum_{i\in I}\alpha_i.T_i) + \sum_{j\in J}\beta_j.R_j)$$

$$A_2 = \tau.S\{A_1/X\} + \sum_{i\in I}\alpha_i.T_i\{A_1/X\}$$

$$B_1 = \mu X(\tau.S + \sum_{i\in I}\alpha_i.T_i + \sum_{j\in J}\beta_j.R_j)$$

First, we show that $A_1 \simeq A_2$. It is obvious that A_1 can simulate A_2. For the other direction, since X is probabilistically unguarded in S, actions $l_i \in \{\tau\} \cup \{p\tau \mid 0 < p < 1\}$ in $A_2 \xrightarrow{\tau} S\{A_1/X\} \xrightarrow{l_1} S_1\{A_1/X\} \xrightarrow{l_2} \ldots \xrightarrow{l_m} A_1$, can be proved state-preserving. We can construct an ϵ-tree $t^{A_2}_{\simeq}$ with every branch stepping into A_1. Thus for every $t^{A_1}_{\simeq}$, there is an ϵ-tree $t^{A_2}_{\simeq}$ with the same set of leaf nodes.

With the fact $A_1 \xrightarrow{\tau} A_2$ is state-preserving, A_1 can simulate B_1. For the other direction, we will construct an ϵ-tree $t^{B_1}_{\simeq}$ by a given $t^{A_1}_{\simeq}$. $t^{B_1}_{\simeq}$ does nothing if $A_1 \xrightarrow{\tau} A_2$ in $t^{A_1}_{\simeq}$, and follows $t^{A_1}_{\simeq}$ in other cases. It can be seen that $t^{A_1}_{\simeq}$ and $t^{B_1}_{\simeq}$ have the same set of leaf nodes.

Corollary 1 (Soundness). *For* $E, F \in \mathcal{P}_{RCCS_{fs}}$, *if* $B1-2$, $R1-4 \vdash E = F$, *then* $E \simeq F$.

4.3 Completeness

By induction on the structure of $\mathcal{T}_{\mathrm{RCCS}_{fs}}$, we can prove that:

Lemma 1. *For a term* $T \in \mathcal{T}_{RCCS_{fs}}$,

- *if* $T \xrightarrow{X} 0$, *then* $\vdash T = X$;
- *if* $T \in \mathcal{T}_d$, *then* $\vdash T = \sum_{i\in I}\{\alpha_i.T_i | T \xrightarrow{\alpha_i} T_i\}$;
- *if* $T \in \mathcal{T}_p$, *then* $\vdash T = \bigoplus_{i\in I}\{p_i\tau.T_i | T \xrightarrow{p_i\tau} T_i\}$.

Definition 9. *A* recursive specification $\mathbb{S}$ *is a set of equations* $\{X = S_X | X \in V_{\mathbb{S}}\}$ *with* $V_{\mathbb{S}}$ *being a variable set. Process* E $\mathcal{A}$-provably satisfies the recursive specification $\mathbb{S}$ in the variable $X_0 \in V_{\mathbb{S}}$ *if there are processes* E_X *for* $X \in V_{\mathbb{S}}$ *with* $E = E_{X_0}$, *such that for* $X \in V_{\mathbb{S}}$

$$\mathcal{A} \vdash E_X = S_X\{E_Y/Y\}_{Y\in V_{\mathbb{S}}}$$

Let $\mathbb{S}$ be a specification, and $X, Y \in V_{\mathbb{S}}$ define $X >_u Y$ if Y occurs free and probabilistically unguarded in E_X. $\mathbb{S}$ is called guarded if $>_u$ is well-founded on $V_{\mathbb{S}}$.

Proposition 10 (Unique Solutions). *If $\mathbb{S}$ is a finite guarded recursive specification and $X_0 \in V_{\mathbb{S}}$, then there is a process E which R1-provably satisfies $\mathbb{S}$ in X_0. Moreover if there are two such processes E and F, then $R2 \vdash E = F$.*

Proof. By induction on the number of equations of $\mathbb{S}$ as in [22].

Proposition 11. *Let $E_0, F_0 \in \mathcal{P}^g_{RCCS_{fs}}$. If $E_0 \simeq F_0$, then there is a finite guarded recursive specification $\mathbb{S}$ provably satisfied in the same variable X_0 by both E_0 and F_0.*

Proof. Take a fresh set of variables $V_{\mathbb{S}} = \{X_{EF} | E \in \mathcal{P}_{E_0}, F \in \mathcal{P}_{F_0}, E \simeq F\}$. $X_0 = X_{E_0F_0}$. Now for $X_{EF} \in V_{\mathbb{S}}$, $\mathbb{S}$ contains the following equations:

1. If $E \in \mathcal{T}_p$, and for every E' such that $E \xrightarrow{p\tau} E'$, $E' \simeq E$, then $X_{EF} = \bigoplus\{p\tau.X_{E'F} | E \xrightarrow{p\tau} E'\}$.
2. If condition in case 1 is not satisfied, $F \in \mathcal{T}_p$, and for every F' such that $F \xrightarrow{p\tau} F'$, $F' \simeq F$, then $X_{EF} = \bigoplus\{p\tau.X_{EF'} | F \xrightarrow{p\tau} F'\}$.
3. If conditions in case 1 and 2 are not satisfied, and $E \in \mathcal{T}_p, F \in \mathcal{T}_p$, then for every $\mathcal{B}_i$ such that $E \rightsquigarrow_{\simeq} \xrightarrow{q} \mathcal{B}_i$, $F \rightsquigarrow_{\simeq} \xrightarrow{q} \mathcal{B}_i$, choose a pair of processes $E_i, F_i \in \mathcal{B}_i$, $X_{EF} = \bigoplus_{\mathcal{B}_i}\{q\tau.X_{E_iF_i}\}$.
4. If $E \in \mathcal{T}_d, F \in \mathcal{T}_d$ then $X_{EF} = \sum\{\alpha.X_{E'F'} | E \xrightarrow{\alpha} E', F \xrightarrow{\alpha} F', E' \simeq F'\} + \sum\{\tau.X_{E'F} | E \xrightarrow{\tau} E', E' \simeq F\} + \sum\{\tau.X_{EF'} | F \xrightarrow{\tau} F', E \simeq F'\}$.
5. Otherwise, $X_{EF} = \sum\{\tau.X_{E'F} | E \xrightarrow{\tau} E', E' \simeq F\} + \sum\{\tau.X_{EF'} | F \xrightarrow{\tau} F', E \simeq F'\}$.

The corresponding process of variable X_{EF} is E. We will prove $B1$–$2, R1$–$2 \vdash E = S_{X_{EF}}\{E'/X_{E'F'}\}_{X_{E'F'} \in V_S}$. Then E_0 is $B1-2, R1-2$ provably satisfying S in X_0. The same statement for F_0 then follows by symmetry.

The case 1, 2, 5 can be proved directly by Lemma 1.

Case 3 is the different part with the proof in [10]. In case 3, $E \in \mathcal{T}_p, F \in \mathcal{T}_p$, and both of E and F can directly do some probabilistic τ action to a different equivalence class. It will be sufficient to prove the following claim:

Claim. For $G \in \mathcal{P}^g_{\mathrm{RCCS}_{fs}}$, if $G \rightsquigarrow_{\simeq} \xrightarrow{q_i} E_i$ for $i \in I$, then $B1-2, R1-2 \vdash G = \bigoplus_{i \in I} q_i\tau.E_i$.

Proof. Define the lexicographic ordering $(m, n) < (m', n')$ as $m < m'$ or ($m = m'$ and $n < n'$). We also define $(m_1, n_1) + (m_2, n_2) = (m_1 + m_2, n_1 + n_2)$ and $(m, n)_1 = m, (m, n)_2 = n$.

Define the following rank function $r : \mathcal{P}^g_{\mathrm{RCCS}_{fs}} \to \mathbb{N} \times \mathbb{N}$:

$$
\begin{aligned}
r(\mathbf{0}) &= (0, 0) \\
r(\bigoplus_{i \in I} p_i\tau.E_i) &= (0, 1) + \max_{i \in I}\{r(E_i)\} \\
r(\sum_{i \in I} \alpha_i.E_i) &= \max\Big\{\{(0, 1) + r(E_i) | \alpha_i = \tau\} \cup \{(0, 1) | \alpha_i \neq \tau\}\Big\} \\
r(\mu X.T) &= (1 + r(T\{\mathbf{0}/X\})_1, 0)
\end{aligned}
$$

By induction on $r(G)$, we can formally prove case 3:

- If $r(G) = (0,1)$, and $G \rightsquigarrow_{\simeq} \xrightarrow{q_i} E_i$ for $i \in I$.
  Then G must be of the form $\bigoplus_{i\in I}\{\bigoplus_{j\in J_i} p_j\tau.E_i | \sum_{j\in J_i} p_j = q_i\}$. Then $(A1) \vdash G = \bigoplus_{i\in I} q_i\tau.E_i$.
- If $r(G) = (m,n) > (0,1)$, and $G \rightsquigarrow_{\simeq} \xrightarrow{q_i} E_i$ for $i \in I$.
  - $G = \sum_{j\in J} \alpha_j.G_j$. For every $j \in J$, $\alpha_j = \tau$ and $G \simeq G_j \rightsquigarrow_{\simeq} \xrightarrow{q_i} E_i$. And $r(G_j) < r(G)$.
    By induction hypothesis, $B1-2, R1-2 \vdash G_j = \bigoplus_{i\in I} q_i\tau.E_i$ for every $j \in J$. We can conclude that $B1-2, R1-2 \vdash G = \bigoplus_{i\in I} q_i\tau.E_i$ for every $j \in J$.
  - $G = \bigoplus_{j\in J} p_j\tau.G_j$. If there exists some $j \in J$, $G_j \not\simeq G$, by Lemma 1, $\vdash G = \bigoplus_{i\in I}\{p_i\tau.E_i | \frac{p_i}{q_i} = c < 1, G \xrightarrow{p_i\tau} E_i \not\simeq G\} \oplus \bigoplus_{j\in J-I}\{p_j\tau.G_j : G \xrightarrow{p_j\tau} G_j \simeq G\}$. For every $j \in J - I$, $r(G_j) < r(G)$. By induction hypothesis, $B1, B2, R2 \vdash G_j = \bigoplus_{i\in I} q_i\tau.E_i$ for every $j \in J - I$, then $B2(A1) \vdash G = \bigoplus_{i\in I} q_i\tau.E_i$.
    If for every $j \in J$, $G_j \simeq G$. Then for every $j \in J$, $G_j \rightsquigarrow_{\simeq} \xrightarrow{q_i} E_i$ and $r(G_j) < r(G)$. By induction hypothesis, $B1$–$2, R1$–$2 \vdash G_j = \bigoplus_{i\in I} q_i\tau.E_i$, then $(A1) \vdash G = \bigoplus_{i\in I} q_i\tau.E_i$.
  - $G = \mu X.T$
    $T\{\mu X.T/X\} \rightsquigarrow_{\simeq} \xrightarrow{q_i} E_i$, then $T\{\bigoplus_{i\in I} q_i\tau.E_i/X\} \rightsquigarrow_{\simeq} \xrightarrow{q_i} E_i$. Since $r(\mu X.T)_1 = 1 + r(T\{\bigoplus_{i\in I} q_i\tau.E_i/X\})_1$, $r(\mu X.T) > r(T\{\bigoplus_{i\in I} q_i\tau.E_i/X\})$.
    By induction hypothesis, $\vdash T\{\bigoplus_{i\in I} q_i\tau.E_i/X\} = \bigoplus_{i\in I} q_i\tau.E_i$, $R2 \vdash \mu X.T = \bigoplus_{i\in I} q_i\tau.E_i$.

In case 4, $E \in \mathcal{T}_d$ and $F \in \mathcal{T}_d$. We need to prove

$$\begin{aligned} B1 \vdash E = &\sum\{\alpha.E' | E \xrightarrow{\alpha} E', F \xrightarrow{\alpha} F', E' \simeq F'\} \\ &+ \sum\{\tau.E' | E \xrightarrow{\tau} E', E' \simeq F\} + \sum\{\tau.E | F \xrightarrow{\tau} F', E \simeq F'\} \end{aligned} \tag{4}$$

By Lemma 1, $\vdash E = \sum_{i\in I}\{\alpha_i.E_i : E \xrightarrow{\alpha_i} E_i\}$, then

$$\begin{aligned} B1 \vdash E = &\sum\{\alpha.E' | E \xrightarrow{\alpha} E', F \xrightarrow{\alpha} F', E' \simeq F'\} \\ &+ \sum\{\tau.E' | E \xrightarrow{\tau} E', E' \simeq F\} + \tau.E \end{aligned} \tag{5}$$

If there exists a process F' with $F \xrightarrow{\tau} F' \simeq E$, (4) and (5) are equal directly. Otherwise, every action from E should be bisimulated by F directly, which means the set $\{\alpha_i.E_i : E \xrightarrow{\alpha_i} E_i\}$ equals to the set $\{\alpha.E' | E \xrightarrow{\alpha} E', F \xrightarrow{\alpha} F', E' \simeq F'\} \cup \{\tau.E' | E \xrightarrow{\tau} E', E' \simeq F\}$.

Corollary 2 (Completeness for probabilistically guarded processes). *For $E, F \in \mathcal{P}^g_{RCCS_{fs}}$, if $E \simeq F$ then $B1-2$, $R1-2 \vdash E = F$.*

Proposition 12. *For $E \in \mathcal{P}_{RCCS_{fs}}$, there exists a probabilistically guarded process E' with $R1,3,4 \vdash E = E'$.*

Proof. Induction on the depth of nesting of recursions in $\mu X.T$ [10].

Corollary 3 (Completeness for all processes). *For E,$F \in \mathcal{P}_{RCCS_{fs}}$, if $E \simeq F$ then $B1-2$, $R1-4 \vdash E = F$.*

5 Concluding Remarks

We have studied algorithm and axiomatization of the branching bisimulation relations for randomized CCS model. We give a polynomial time algorithm for equivalence checking and show that our axiom system is sound and complete. These two results, besides their value to the randomized CCS model itself, can be generalized to other randomized finite state models. The reason is that the essence of our work is dealing with probabilistic actions, which however, is model independent.

We are currently planning to extend our axiomatization to the divergence-sensing branching bisimulation and other equivalences such as testing equivalence. Another interesting topic is to implement the ϵ-tree technique on other classical probabilistic process calculi. We believe this is an expecting topic as it can be regarded as an extension and application of the philosophy of the model independent method.

Acknowledgement. We are grateful to Prof. Yuxi Fu for his instructive discussions and feedbacks. We thank Dr. Mingzhang Huang, Dr. Qiang Yin and other members of BASICS for offering helps in the revision stage. We also thank the anonymous referees for their questions and detailed comments. The support from the National Science Foundation of China (61772336, 61872142, 61572318) is acknowledged.

References

1. Andova, S.: Process algebra with probabilistic choice. In: Katoen, J.-P. (ed.) ARTS 1999. LNCS, vol. 1601, pp. 111–129. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48778-6_7
2. Baeten, J.C.M., Bergstra, J.A., Smolka, S.A.: Axiomatizing probabilistic processes: Acp with generative probabilities. Inf. Comput. **121**(2), 234–255 (1995)
3. Baier, C., Hermanns, H.: Weak bisimulation for fully probabilistic processes. In: Grumberg, O. (ed.) CAV 1997. LNCS, vol. 1254, pp. 119–130. Springer, Heidelberg (1997). https://doi.org/10.1007/3-540-63166-6_14
4. Bandini, E., Segala, R.: Axiomatizations for probabilistic bisimulation. In: Orejas, F., Spirakis, P.G., van Leeuwen, J. (eds.) ICALP 2001. LNCS, vol. 2076, pp. 370–381. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-48224-5_31
5. Deng, Y.: Semantics of Probabilistic Processes: An Operational Approach. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-45198-4
6. Deng, Y., Palamidessi, C.: Axiomatizations for probabilistic finite-state behaviors. In: Sassone, V. (ed.) FoSSaCS 2005. LNCS, vol. 3441, pp. 110–124. Springer, Heidelberg (2005). https://doi.org/10.1007/978-3-540-31982-5_7

7. Fu, Y.: Checking equality and regularity for normed BPA with silent moves. In: Fomin, F.V., Freivalds, R., Kwiatkowska, M., Peleg, D. (eds.) ICALP 2013, Part II. LNCS, vol. 7966, pp. 238–249. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39212-2_23
8. Fu, Y.: A uniform approach to random process model (2019). https://arxiv.org/pdf/1906.09541.pdf
9. Giacalone, A., Jou, C.C., Smolka, S.A.: Algebraic reasoning for probabilistic concurrent systems. In: Proceedings of IFIP TC2 Working Conference on Programming Concepts and Methods. Citeseer (1990)
10. Glabbeek, R.J.: A complete axiomatization for branching bisimulation congruence of finite-state behaviours. In: Borzyszkowski, A.M., Sokołowski, S. (eds.) MFCS 1993. LNCS, vol. 711, pp. 473–484. Springer, Heidelberg (1993). https://doi.org/10.1007/3-540-57182-5_39
11. van Glabbeek, R.J., Smolka, S.A., Steffen, B.: Reactive, generative, and stratified models of probabilistic processes. Inf. Comput. **121**(1), 59–80 (1995)
12. van Glabbeek, R.J., Weijland, W.P.: Branching time and abstraction in bisimulation semantics. J. ACM **43**(3), 555–600 (1996)
13. Hansson, H., Jonsson, B.: A framework for reasoning about time and reliability. In: Proceedings of Real-Time Systems Symposium, pp. 102–111. IEEE (1989)
14. Hansson, H., Jonsson, B.: A calculus for communicating systems with time and probabilities. In: Proceedings of 11th Real-Time Systems Symposium, pp. 278–287. IEEE (1990)
15. Herescu, O.M., Palamidessi, C.: Probabilistic asynchronous π-calculus. In: Tiuryn, J. (ed.) FoSSaCS 2000. LNCS, vol. 1784, pp. 146–160. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-46432-8_10
16. Huang, M., Yin, Q.: Two lower bounds for BPA. In: 28th International Conference on Concurrency Theory, CONCUR 2017, 5–8 September 2017, Berlin, Germany, pp. 20:1–20:16 (2017). https://doi.org/10.4230/LIPIcs.CONCUR.2017.20
17. Jou, C.-C., Smolka, S.A.: Equivalences, congruences, and complete axiomatizations for probabilistic processes. In: Baeten, J.C.M., Klop, J.W. (eds.) CONCUR 1990. LNCS, vol. 458, pp. 367–383. Springer, Heidelberg (1990). https://doi.org/10.1007/BFb0039071
18. Kučera, A., Jančar, P.: Equivalence-checking on infinite-state systems: techniques and results. Theory Pract. Logic Programm. **6**(3), 227–264 (2006)
19. Larsen, K.G., Skou, A.: Bisimulation through probabilistic testing. Inf. Comput. **94**(1), 1–28 (1991)
20. Lowe, G.: Probabilities and priorities in timed CSP (1993)
21. Milner, R.: Communication and Concurrency, vol. 84. Prentice hall, New York (1989)
22. Milner, R.: A complete axiomatisation for observational congruence of finite-state behaviours. Inf. Comput. **81**(2), 227–247 (1989)
23. Philippou, A., Lee, I., Sokolsky, O.: Weak bisimulation for probabilistic systems. In: Palamidessi, C. (ed.) CONCUR 2000. LNCS, vol. 1877, pp. 334–349. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-44618-4_25
24. Stark, E.W., Smolka, S.A.: A complete axiom system for finite-state probabilistic processes. In: Proof, Language, and Interaction, pp. 571–596 (2000)