Research Article
Quantitative Analysis of Software Approximate Correctness

Yanfang Ma

School of Computer Science and Technology, Huaibei Normal University, Huaibei 235000, China

Correspondence should be addressed to Yanfang Ma; myf1978@gmail.com

Received 6 November 2014; Accepted 14 June 2015

Academic Editor: Fabio Tramontana

Copyright © 2015 Yanfang Ma. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Parameterized bisimulation provides an abstract description of software correctness. In real world situations, however, many software products are approximately correct. To characterize the approximate correctness, we generalize the parameterized bisimulation to numerical version and probabilistic setting. First, we propose the definition of the parameterized bisimulation index that expresses the degree to which a binary relation is parameterized bisimulation. Then, $\lambda$-parameterized bisimulation over environment $\varepsilon$ and its substitutivity laws are presented. Finally, $\lambda$-parameterized probabilistic bisimulation is established to describe complicated software products with probabilistic phenomena.

1. Introduction

Correctness is a key feature of software trustworthiness [1–3], which can be abstracted by using various behavior equivalences between processes, such as (strong and weak) bisimilarity, trace equivalence, testing equivalence, and failure equivalence [4–6]. Specification and implementation of software are considered as two processes. If a certain behavior equivalence exists between specification and implementation, then the software is considered as correctness. Thus a certain behavior equivalence must be established between specification and implementation to prove software correctness.

However, the prerequisites for successful application of software products may not always hold when they are actually running on the computers. As physical devices, computers cannot be assumed to behave reliably. In addition, standard implementations at best approximate the formal definition of semantics. Ying [7] proposed strong/weak bisimulation indexes to establish the approximate description between specification and implementation. The proposed indexes characterize the degree to which a binary relation between processes is strong/weak bisimulations. Ying and Wirsing [8] presented the strong/weak bisimulation limits and obtained the strong/weak bisimulation topologies to describe that the sequence of implementations can be treated as an evolution toward the specification. Girard and Pappas [9] defined a hierarchy of approximate pseudometric between two systems that quantifies the qualities of the approximations. To verify whether a program behaved as desired, Henzinger [10] introduced quantitative fitness measures for programs, particularly to measure the function, performance, and robustness of reactive programs such as concurrent processes. To compare these existing quantitative models of program approximate correctness, Fahrenberg and Legay [11] presented a distance-agnostic approach to quantify the verification. They defined a spectrum of different interesting system distances that corresponds to the given trace distance.

In fact, some complicated software products contain probabilistic phenomena. These software products can be abstracted as probabilistic processes. Similarly, many quantitative models based on probabilistic processes [12] have existed to obtain the degree to which implementations satisfy their specification. For example, Giacalone et al. [13, 14] presented $\varepsilon$-bisimulation equivalence relation over deterministic probabilistic processes and proposed a kind of measure model to describe the degree of similarity among probabilistic processes. This measure is defined based on the probability differences of the processes that execute the same action. Song et al. [15] proposed a measure model according to the probability of the processes that performs the same trace with a discount factor. Deng et al. [16] defined
state-metrics as a natural extension of bisimulation from nonquantitative systems to quantitative ones over action-labeled quantitative systems. Alves de Medeiros et al. [17] built a measure relation based on the observable actions of processes. Abate [18] also established an approximate metric based on probabilistic bisimulation.

However, the running of a software depends on its environment. The environment should be considered when the approximation degree between specification and implementation is discussed. The influences of the environment are absent in the existing quantitative models. In [19], Larsen and Skou presented two-thirds bisimulation based on probabilistic transition systems to characterize that two processes are undistinguished when they have the same sets of observations for all tests. If an environment is considered as a set of actions [20], then two-thirds bisimulation expresses the relation in which the process refuses the environment. We proposed two-thirds simulation index and established a measure model to describe the degree of approximation among processes [21]. In \( \pi \)-calculus [22] and applied \( \pi \)-calculus [23], the observation equivalences were researched. And the influence of environment on the execution of software was considered as well. In [22], a process context is speaking a process expression containing a hole. In [23], the contexts may be used to represent the adversarial environment in which a process is run. The environment provides the data that the process inputs and consumes the data that it outputs.

Larsen [24] presented parameterized bisimulation equivalence to obtain flexible hierarchic development methods. In the work of Larsen, bisimulation equivalence is parameterized with information about context called environment. Environment \( e \) is considered as an object that consumes the actions produced by a process in that environment. However, the abilities of environment to consume actions might be limited. Suppose that \( P \) is a process, and it can execute action \( \alpha \) to next process \( P' \); that is, \( P \xrightarrow{\alpha} P' \). However, \( e \) cannot consume the action \( \alpha \); then derivation \( P \xrightarrow{\alpha} P' \) will never be considered when \( P \) is executed in environment \( e \). If \( P \) and \( Q \) both perform the same action for all transitions of \( e \), then we can determine \( P \xrightarrow{\alpha} Q \). In particular, strong bisimulation in CCS (Communication and Concurrency Systems) model is generalized by parameterized bisimulation equivalence. Parameterized limit bisimulation and parameterized bisimulation limit were proposed in [25, 26] to describe the infinite evolution mechanism.

The conditions possessing the same observable actions consumed by the environment are rigorous when we choose parameterized bisimulation to verify software correctness. Sometimes we can determine that two processes fail to meet these conditions. However, these processes are still close to parameterized bisimulation in the sense that whenever a process can execute an action of environment consumption, another process can produce an action that is different from the environment consuming.

The aim of this study is to build mathematical tools that are suitable for describing this kind of approximate parameterized bisimulation. First, we propose parameterized bisimulation index over environment \( e \) in order to describe the degree to which binary relation \( R_e \) is a parameterized bisimulation. Then we define \( \lambda \)-parameterized bisimulation and discuss algebraic properties. We specially prove the congruence of \( \lambda \)-parameterized bisimulation under various operators. Finally, in order to describe the characterization of software with probabilistic information, we also extend parameterized bisimulation to probabilistic setting and propose the approximate parameterized probabilistic bisimulation.

Compared with the main focuses of [7, 8, 21], the main focus of our work is on parameterized bisimulation. In [7, 8], the set of labels in a labeled transition system is equipped with a metric. Given a binary relation between processes, the degree to which the relation is bisimulation is defined. Similar to [7, 21], we also equip the set of actions with a metric. Parameterized bisimulation that includes the information about context is different from bisimulation. For every environment \( e \), \( R_e \) is a binary relation between processes. Therefore, in order to obtain the approximate parameterized bisimulation, we need to establish the bisimulation index for every environment \( e \). We consider two cases to obtain the bisimulation index for an environment. One case is that when the environment consumes an action, a process can accept this action and another process cannot accept this action. Another case is that when the environment has the transition with an action, two processes cannot both accept this action. Therefore, our definition about bisimulation index on the environment is different from the definition of bisimulation index in [7]. Furthermore, we establish the \( \lambda \)-parameterized bisimulation on the environment \( e \). In order to obtain the hierarchic development and modular decomposition of software, similar to [7, 21], we also consider the substitutivity laws of \( \lambda \)-parameterized bisimulation on the environment under various combinators.

Meanwhile, we notice that many metric models are proposed based on the difference of probabilities in which two processes execute the same action [16]. But the influence of environment was not considered in these models. In order to describe the approximation of the complicated software with probabilistic information, we extend parameterized bisimulation to probabilistic setting in order to reflect the environment. First, we extend the environment transition system to probabilistic case. Then, we define parameterized probabilistic bisimulation. Finally, we obtain the \( \lambda \)-parameterized probabilistic bisimulation based on the probabilities that the environment consumes an action and the processes perform the same action. This point is similar to [14, 16]. Our method is different from the method in [18]. In [18], the state space is equipped with a rich structure, whereas the metric is characterized by probabilistic conditional kernels.

In Section 2, we recall the syntax of CCS and parameterized bisimulation. Parameterized bisimulation index over environment \( e \) and \( \lambda \)-parameterized bisimulation are
defined in Section 3. Their some algebraic properties are researched in Section 3. In Section 4, the substitutivity laws of \( \lambda \)-parameterized bisimulation under various operators are proved. In Section 5, parameterized probabilistic bisimulation is proposed and \( \lambda \)-parameterized probabilistic bisimulation is defined. Furthermore, the congruence of \( \lambda \)-parameterized probabilistic bisimulation is proved. Our conclusions and future work are presented in Section 6.

2. Preliminaries

2.1. CCS Summary. This section recalls some fundamental concepts and the results of process calculus needed in the subsequent sections. The following definitions mainly come from the book by Ying [27].

We introduce the names \( \mathcal{A} \), the conames \( \bar{\mathcal{A}} \), and labels \( \Gamma = \mathcal{A} \cup \bar{\mathcal{A}}, a, b, \ldots \) range over \( \mathcal{A}, \bar{\mathcal{A}}, b, \ldots \) range over \( \mathcal{A}, \bar{\mathcal{A}}, l, \ldots \) range over \( \Gamma \). We also introduce the silent or perfect action \( \tau \). \( \text{Act} = \Gamma \cup \{ \tau \} \) is defined as the set of actions, whereas \( \alpha, \beta \) are defined range over \( \text{Act} \). Furthermore, we introduce the set \( \mathcal{N} \) of process variables and set \( \mathcal{K} \) of processes constants. Mapping \( f : \Gamma \rightarrow \Gamma \) is a relabeling function if \( f(l) = l \) for every \( l \in \Gamma \). We may extend relabeling function \( f \) to be a mapping from \( \text{Act} \) to itself by decreeing that \( f(\tau) = \tau \). The syntax of the basic process calculus is presented in the following definition.

**Definition 1** (process expression [28]). The class \( \partial \) of process expressions is the smallest class of symbol strings that satisfies the following conditions:

1. \( \mathcal{N}, \mathcal{K} \in \partial \).
2. If \( \alpha \in \text{Act} \) and \( E \in \partial \), then \( \alpha, E \in \partial \).
3. If \( I \) is an indexing set and \( E_i \in \partial \) \( \forall i \in I \), then \( \sum_{i \in I} E_i \in \partial \).
4. If \( E_1, E_2 \in \partial \), then \( E_1 \mid E_2 \in \partial \).
5. If \( E \in \partial \) and \( L \subseteq \Gamma \), then \( E \mid L \in \partial \).
6. If \( E \in \partial \) and \( f \) is a relabeling function, then \( E[f] \in \partial \).

The process expressions without process variables are called processes and the class of processes is denoted by \( \mathcal{P} \). For any \( A \in \mathcal{N} \), we assume that there is a defining equation \( A \overset{\text{def}}{=} P_A \), such as \( A = P_A = 1 \cdot A \in \mathcal{P} \). Constants provide us a mechanism of recursion in the process calculus.

The transitional semantics of the basic calculus is presented in the style of Plotkin’s structural operational semantics [29]. We have the following definition.

**Definition 2** (labeled transition system [7]). Let \((\partial, \text{Act}, \overset{\alpha}{\rightarrow})\) be a labeled transition system, where the transition relations \( \overset{\alpha}{\rightarrow} \) (\( \alpha \in \text{Act} \)) are presented by the following rules:

\[
\begin{align*}
\overset{\alpha}{\rightarrow} & : A \rightarrow A, \\
E & \overset{\alpha}{\rightarrow} E' \\
\sum_{i \in I} E_i & \overset{\alpha}{\rightarrow} E_j' \quad (j \in I),
\end{align*}
\]

\[
\begin{align*}
\text{Com}_1 & : E \overset{\alpha}{\rightarrow} E' \quad E \mid F \overset{\alpha}{\rightarrow} E' \mid F \\
\text{Com}_2 & : F \overset{\alpha}{\rightarrow} F' \quad E \mid F \overset{\alpha}{\rightarrow} E' \mid F \\
\text{Com}_3 & : E \overset{\tau}{\rightarrow} E' \quad F \overset{\tau}{\rightarrow} F' \\
\text{Res} & : E \overset{\alpha}{\rightarrow} E' \quad (\alpha, \bar{\alpha} \notin L), \\
\text{Rel} & : E \overset{\alpha}{\rightarrow} E' \quad E[f] \overset{\alpha}{\rightarrow} E'[f], \\
\text{Con} & : P \overset{\alpha}{\rightarrow} P' \quad \left( A \overset{\text{def}}{=} P \right).
\end{align*}
\]

Transitions with strings of labels may be defined in a natural way. If \( t = \alpha_1, \ldots, \alpha_n \in \text{Act}^* = \bigcup_{n \geq 0} \text{Act}^n \), then we write \( E \overset{t}{\rightarrow} E' \) provided that \( E \overset{\alpha_i}{\rightarrow} E_{i+1} \) for some \( E_i, \ldots, E_{n-1} \in \mathcal{P} \). In this case, we call \( t \) an action sequence of \( E \) and \( E' \) is a \( t \)-derivative of \( E \). If for some \( t \in \text{Act}^* \), \( E' \) is a \( t \)-derivative of \( E \), then \( E' \) is called a derivative of \( E \).

In the subsequent sections, we mainly consider the restriction \((\mathcal{P}, \text{Act}, \overset{\alpha}{\rightarrow})|_{\mathcal{P}}\) of \((\mathcal{P}, \text{Act}, \overset{\alpha}{\rightarrow})\) on \( \mathcal{P} \), where, for each \( \alpha \in \text{Act} \), \( \overset{\alpha}{\rightarrow} \mid_{\mathcal{P}} = \overset{\alpha}{\rightarrow} \cap (\mathcal{P} \times \mathcal{P}) \) is restriction of \( \overset{\alpha}{\rightarrow} \) on \( \mathcal{P} \). For simplicity, we always write \( \overset{\alpha}{\rightarrow} \) for \( \overset{\alpha}{\rightarrow} \mid_{\mathcal{P}} \).

For example, suppose that a vending machine that sells CocaCola can be described as an expression of CCS:

\[
P = 1d.\text{CocaCola}.\text{collect}.P.
\]

(2)

Its behavior can be expressed as a transition diagram as in Figure 1.

2.2. Parameterized Bisimulation. The definition of environment must be introduced because the motivation of parameterized bisimulation is to parameterize the bisimulation equivalence with a special type of information about context called environment. Similar to the assumption that a process may change after performing an action, the assumption that an environment may change after consuming an action is reasonable. Thus environments and their behaviors can be described by labeled transition system \( e = (\text{Env}, \text{Act}, \Rightarrow) \), where Env is the set of environments, Act is the set of actions (identical to the set of actions used in the transition system of process), and \( \Rightarrow \) is a subset of \( \text{Env} \times \text{Act} \times \text{Env} \) called consumption relation. \( e \overset{\alpha}{\Rightarrow} e' \) means "\( e \) may consume the action \( \alpha \) and in doing so become the environment \( e' \)."
At this point, let us review the parameterized bisimulation equivalence. First, we recall the bisimulation equivalence without environment.

**Definition 3** (bisimulation [24]). Bisimulation $R$ is a binary relation on $\mathcal{P}$ such that whenever $PRQ$ and $\alpha \in \text{Act}$, then

1. $P \xrightarrow{\alpha} P' \Rightarrow \exists Q' \text{ such that } Q \xrightarrow{\alpha} Q' \text{ and } P' R Q'$,
2. $Q \xrightarrow{\alpha} Q' \Rightarrow \exists P' \text{ such that } P \xrightarrow{\alpha} P' \text{ and } P' R Q'$.

Two processes, $P$ and $Q$, are considered bisimulation if and only if bisimulation $R$ exists and satisfies $(P, Q) \in R$.

**Definition 4** ($\varepsilon$-parameterized bisimulation [24]). Let $\varepsilon = (\text{Env}, \text{Act})$ be a transition system of environments. Then an $\varepsilon$-parameterized bisimulation, $R$, is an Env-indexed family of binary relations, $R_\varepsilon \subseteq \mathcal{P} \times \mathcal{P}$ for $\varepsilon \in \text{Env}$, s.t. whenever $PR_\varepsilon Q$ and $e \xrightarrow{\alpha} f$, then we have the following:

1. If $P \xrightarrow{\alpha} P'$, then there exists $Q'$, s.t. $Q \xrightarrow{\alpha} Q'$ and $P' R_f Q'$.
2. If $Q \xrightarrow{\alpha} Q'$, then there exists $P'$, s.t. $P \xrightarrow{\alpha} P'$ and $P' R_f Q'$.

Two processes, $P$ and $Q$, are said to be bisimulation equivalence in the environment $e \in \text{Env}$ if and only if $\varepsilon$-parameterized bisimulation $R$ exists, such that $PR_\varepsilon Q$, which is denoted by $P \sim_\varepsilon Q$.

**Example 5** (see [24]). Let $e$, $P$, and $Q$ be presented by Figure 2. The Env-indexed family $R$ is shown as follows:

$$R_{e_2} = \{(P_0, Q_0)\}.$$  
$$R_{e_1} = \{(P_1, Q_1), (P_2, Q_2), (P_1, Q_3)\}.$$  

We can prove that $R$ is a parameterized bisimulation. Thus, $P_0 \sim_{e_2} Q_0$, $P_0 + Q_0$. Therefore, $P_0$ can accept the action $a$, $Q_0$ can also accept the same action, and their next states have the relation $R_{e_2}$ when the environment $e_0$ can consume the action $a$ to the next environment $e_1$. By contrast, if $Q_0$ can accept the action $a$, then $P_0$ can also accept the same action, and their next states have the relation $R_{e_1}$. Similarly, $P_0$ and $Q_0$ have the same behavior when environment $e_0$ can consume action $a$ to the next environment $e_2$. Thus, $P_0 \sim_{e_2} Q_0$. However, according to Definition 3, we can observe that $P_0 + Q_0$.

Although $(P_0, Q_0) \in R_{e_2}, P_2$ and $Q_2$ will never be considered when $P_2$ and $Q_2$ are executed in environment $e_1$. The reason is that when $e_1$ consumes the action $b$, the environment $e_3$, $P_2$ and $Q_2$ cannot execute the action $b$ to the next state.

**Proposition 6** (see [24]). For all $e \in \text{Env}$ and for all $P, Q \in \mathcal{P}$, $P \sim Q$ implies that $P \sim_\varepsilon Q$.

This proposition indicates that parameterized bisimulation equivalence generalizes bisimulation equivalence.

### 3. Approximate Parameterized Bisimulation

For the approximate version of parameterized bisimulation, we present the definition of parameterized bisimulation index over the environment that indicates the degree to which a binary relation is parameterized bisimulation. We also generalize some algebraic properties of parameterized bisimulation.

**Definition 7** (metric space [30]). Let $M$ be a nonempty set. $\rho$ is a mapping from $M \times M$ into $[0, \infty]$. Then the pair $(M, \rho)$ is called a metric space if the following conditions are satisfied:

1. $\rho(x, y) = 0$ if and only if $x = y$.
2. $\rho(x, y) = \rho(y, x)$.
3. $\rho(x, z) \leq \rho(x, y) + \rho(y, z)$ for any $x, y, z \in M$.

If (1) is weakened by (1)′: $\rho(x, x) = 0$ for each $x \in M$, then $\rho$ is called a pseudometric. If (3) is strengthened by (3)′:
\(\rho(x, z) \leq \max(\rho(x, y), \rho(y, z))\) for any \(x, y, z \in M\), then \(\rho\) is called an ultrametric.

Let \(\rho\) be a metric on \(\mathcal{A}\). As expected, we can extend \(\rho\) to a mapping from \(\text{Act} \times \text{Act}\) to \([0, \infty]\), which is denoted by \(\rho_{\text{act}}\) in the following way: for any \(a, b \in \mathcal{A}\),

1. \(\rho_{\text{act}}((a, b) = 0\),
2. \(\rho_{\text{act}}(a) = \rho_{\text{act}}(\overline{a}, \overline{b}) = \rho(a, b)\),
3. \(\rho_{\text{act}}(a, \tau) = \rho_{\text{act}}((a, \tau) = \rho_{\text{act}}(\overline{a}, \tau) = \rho_{\text{act}}(\overline{a}, \overline{b}) = \rho(a, b)\).

\(\rho_{\text{act}}\) is clearly a metric on \(\text{Act}\). In addition, \(\rho_{\text{act}}\) is also an ultrametric provided that \(\rho\) is an ultrametric. For simplicity, we always write \(\rho\) for \(\rho_{\text{act}}\). Then the numerical generalization of Definition 4 will be defined. Similar to the parameterized bisimulation, the following assumption is obtained: if \(e_1\) can consume an action to \(e_2\), \(P\) and \(Q\) have the relation \(R_e\), but no transitions exist that can make \(P\) and \(Q\) execute certain actions to obtain some states that are included in \(R_e\), then \(P\) and \(Q\) will never be considered when \(P\) and \(Q\) are executed in \(e_1\).

**Definition 8.** Let \((\mathcal{P}, \text{Act}, \frac{\alpha}{\beta} : \alpha \in \text{Act})\) be a labeled transition system and let \(\rho\) be a metric on \(\text{Act}\). \(\varepsilon = (\text{Env}, \text{Act}, \Rightarrow)\) is an environment transition system. \(R\) is an \(\text{Env-indexed}\) family of binary relations \(R_e \subseteq \mathcal{P} \times \mathcal{P}\). For \(e \in \text{Env}, e \Rightarrow e'\), and \((P, Q) \in R_e\), we define that

\[
\frac{\rho_{(\alpha, e)}}{\rho_{(\alpha, e)}}(P, Q) = \max\left\{\frac{b_{(\alpha, e)}}{b_{(\alpha, e)}}(P, Q) : Q' \in \mathcal{P} \text{ and } t \in \text{Act with } Q \Rightarrow Q', \text{ if } t \neq \alpha\right\},
\]

where

\[
\frac{b_{(\alpha, e)}}{b_{(\alpha, e)}}(P, Q) = \sup\left\{\frac{b_{(\alpha, e)}}{b_{(\alpha, e)}}(P, Q, t) : Q' \in \mathcal{P} \text{ and } t \in \text{Act with } Q \Rightarrow Q', \text{ if } t \neq \alpha\right\},
\]

\[
\frac{\rho_{(\alpha, e)}}{\rho_{(\alpha, e)}}(P, Q) = \max\left\{\frac{b_{(\alpha, e)}}{b_{(\alpha, e)}}(P, Q, t) : Q' \in \mathcal{P} \text{ and } t \in \text{Act with } Q \Rightarrow Q', \text{ if } t = \alpha\right\},
\]

\[
\frac{b_{(\alpha, e)}}{b_{(\alpha, e)}}(P, Q, t) = \inf\left\{\rho(u, a) - \rho(t, a) : P' \in \mathcal{P} \text{ such that } P \stackrel{u}{\rightarrow} P' \text{ and } P' \in \mathcal{P} \text{ and } t \neq \alpha\right\},
\]

\[
\frac{b_{(\alpha, e)}}{b_{(\alpha, e)}}(P, Q, t) = \inf\left\{\rho(u, a) : P' \in \mathcal{P} \text{ such that } P \stackrel{u}{\rightarrow} P', \text{ and } Q' R_{e} P', \text{ if } t = \alpha\right\}.
\]

We call \(b_{(\alpha, e)}(P, Q)\) an index in which \(P\) simulates \(Q\) on the transition \(e \Rightarrow e'\).

**Definition 9.** Let \((\mathcal{P}, \text{Act}, \frac{\alpha}{\beta} : \alpha \in \text{Act})\) be a labeled transition system and let \(\rho\) be a metric on \(\text{Act}\). \(\varepsilon = (\text{Env}, \text{Act}, \Rightarrow)\) is an environment transition system. \(R\) is an \(\text{Env-indexed}\) family of binary relations \(R_e \subseteq \mathcal{P} \times \mathcal{P}\). If \(e \in \text{Env}, \text{ such that} \)

\[
b_{(\alpha, e)} = \sup\left\{b_{(\alpha, e)} : e \Rightarrow e'\right\},
\]

where

\[
b_{(\alpha, e)} = \sup\left\{\max\left\{b_{(\alpha, e)}(P, Q), b_{(\alpha, e)}(Q, P) : PR_{e}Q\right\}\right\},
\]

then \(b_{(\alpha, e)}\) is called parameterized bisimulation index of \(R\) over environment \(e\).

As expected, if \(P, Q', P \in \mathcal{P}, t \in \text{Act} \text{ and } e \Rightarrow e'\) given, and \(PR_{e}Q\) and \(Q' \Rightarrow Q', \text{ then } b_{(\alpha, e)}(P, Q', t)\) and \(b_{(\alpha, e)}(P, Q')\) are the infimum of distances between transitions \(t\) and \(u\) where \(t\) and \(u\) are close to \(\alpha\). From this point, **Definition 9** is clearly a numerical counterpart of Definition 4 and \(b_{(\alpha, e)}\) expresses the degree to which \(R_e\) is parameterized bisimulation. We should indicate that the smaller the value of \(b_{(\alpha, e)}\), the higher the degree to which \(R_e\) is a bisimulation. We can obtain the conclusion that, for every \(e \in \text{Env}, b_{(\alpha, e)} = 0\) when \(R\) is parameterized bisimulation.

**Proposition 10.** (1) \(R\) is parameterized bisimulation if and only if for every \(e \in \text{Env}, b_{(\alpha, e)} = 0\). In particular, \(b_{(\alpha, e)} = 0\), where \(\text{Id}_{(\alpha, e)}\) is the identical relation between processes.

(2) For all \(e \in \text{Env}, b_{(\alpha, e)} = b_{(\alpha, e)}\).

(3) For all \(e \in \text{Env}, b_{(\alpha, e)} \leq b_{(\alpha, e)} + b_{(\alpha, e)}\). In particular, if \(\rho\) is an ultrametric, then \(b_{(\alpha, e)} \leq b_{(\alpha, e)}\).

(4) For all \(e \in \text{Env}, b_{(\alpha, e)} \geq \inf b_{(\alpha, e)}\).

**Proof.** (1) and (2) are direct from Definition 9.

(3) If \(b_{(\alpha, e)} = b_{(\alpha, e)} = \infty\), then it is clear. At this point, suppose that \(b_{(\alpha, e)} = b_{(\alpha, e)} < \infty\). Sequences \(\lambda_{(\alpha, e)}\) and \(\lambda_{(\alpha, e)}\) exist, such that \(b_{(\alpha, e)} < \lambda_{(\alpha, e)}\) and \(b_{(\alpha, e)} < \lambda_{(\alpha, e)}\) (\(n = 1, 2, \ldots\)),

\[
\lim_{n \rightarrow \infty} \lambda_{(\alpha, e)} = b_{(\alpha, e)}, \quad \lim_{n \rightarrow \infty} \lambda_{(\alpha, e)} = b_{(\alpha, e)} \text{ for all } e \Rightarrow e',
\]

\[
b_{(\alpha, e)} \leq \lambda_{(\alpha, e)}, \quad b_{(\alpha, e)} \leq \lambda_{(\alpha, e)}.
\]

For any \(P, W \in \mathcal{P}, \text{ if } P(R_{e}^{1}R_{e}^{2})W, \text{ then there exists } Q \in \mathcal{P}\)

\[
\text{with } PR_{e}^{1}Q \text{ and } QR_{e}^{1}W. \text{ For any } P' \in \mathcal{P} \text{ and } t_{1} \in \text{Act}, P \stackrel{t_{1}}{\rightarrow} P'. \text{ If } t_{1} = \alpha, \text{ then } b_{(\alpha, e)} < \lambda_{(\alpha, e)} \text{ leads to the idea that there exist } Q' \in \mathcal{P} \text{ and } t_{2} \in \text{Act such that} \)

\[
Q' \Rightarrow Q', \text{ and } P' R_{e}^{1}Q', \text{ and } \rho(t_{1}, t_{2}) \leq \lambda_{(\alpha, e)} \text{ at the same time, if } t_{1} \neq \alpha, b_{(\alpha, e)} < \lambda_{(\alpha, e)} \text{ also leads to the idea that there exist } Q' \in \mathcal{P} \text{ and } t_{2} \in \text{Act such that} \)

\[
Q' \Rightarrow Q' \text{ with } \rho(t_{2}, t_{2}) \leq \rho(t_{1}, t_{2}), P' R_{e}^{1}Q', \text{ and } \rho(t_{1}, t_{2}) \leq \lambda_{(\alpha, e)}.\]
Furthermore, $b_{\varepsilon}^n < \lambda_{2n}$ leads to the following: if $t_1 = a$, $b_{\varepsilon}^n(W, Q; t_2) \leq \lambda_{2n}$. Thus, $W' \in \mathcal{P}$ and $t_3 \in \text{Act}$ exist, such that $W \xrightarrow{t_1} W'$ with $\rho(t_2, t_3) < \lambda_{2n}$ and $Q'R_{\varepsilon}^3 W'$. Moreover, if $t_1 \neq a$, $b_{\varepsilon}^n(W, Q; t_2) \leq \lambda_{2n}$. We can obtain $W' \in \mathcal{P}$ and $t_3 \in \text{Act}$ such that $W \xrightarrow{t_1} W'$ with $\rho(t_2, t_3) < \lambda_{2n}$. Therefore, $P(R_{\varepsilon}^3 o R_{\varepsilon}^2) W'$ with $\rho(t_2, t_3) = \rho(t_2, t_3) + \rho(t_2, t_3) < \lambda_{2n}$. By Proposition 12, we should obtain $b_{\varepsilon}^n(W, Q; t_2) < \lambda_{2n}$, and $b_{\varepsilon}^n \leq \lambda_{2n}$. Thus, $b_{\varepsilon}^n \leq \lambda_{2n}$. Finally, we get $b_{\varepsilon}^n(W, Q; t_2) = b_{\varepsilon}^n + b_{\varepsilon}^n$. (4) is similar to (3).

(1) in Proposition 10 indicates that, for any $e \in \text{Env}$, the parameterized bisimulation index is 0, which is the least value of the bisimulation index over the environment $e$. (2) states that, for any environment $e \in \text{Env}$, the bisimulation index of relation $R_{\varepsilon}$ and the bisimulation index of its inverse are the same. (3) means that, for any environment $e \in \text{Env}$, the bisimulation index of the composition of two relations is not greater than the sum of the bisimulation indexes of the relation. If the presumed metric on actions is an ultrametric, then it does not exceed even the greatest of the bisimulation indexes of the factor relations. Finally, (4) means that if the degree to which $R_{\varepsilon}$ is a bisimulation is not less than some values for all $i \in I$, then the degree to which $\bigcup_{i \in I} R_{\varepsilon}$ is a bisimulation is also not less than that value.

Example II. Let $e$ and $P, Q$ be illustrated by Figure 3. The Env-indexed family $R$ is shown as follows:

$$R_{e_0} = \{(P_0, Q_0)\}$$

$$R_{e_1} = \{(P_1, Q_1), (P_2, Q_2), (P_1, Q_3)\}$$

$$R_{e_2} = \{(P_2, Q_2), (P_1, Q_1), (P_2, Q_3)\}$$

$$R_{e_3} = \{(P_3, Q_3), (P_1, Q_1), (P_2, Q_3)\}$$

$$R_{e_4} = \{(P_4, Q_2), (P_4, Q_4)\}$$

Let the metric on Act be defined as:

$$\rho(b, c) = \rho(c, b) = 0.7,$$

$$\rho(b, d) = \rho(d, b) = 0.4,$$

$$\rho(c, d) = \rho(d, c) = 0.8.$$
is $\lambda_i$-parameterized bisimulation ($i \in I$), then $R_i$ is a $\inf_{i \in I} \lambda_i$-parameterized bisimulation.

**Corollary 14.** Let $\sigma = (\mathcal{P}, \text{Act}, (\lambda_\alpha; \alpha \in \text{Act}))$ be a labeled transition system. $e = (\text{Env}, \text{Act}, \Rightarrow)$ is an environment transition system. $R$ is an Env-indexed family of binary relations on $\mathcal{P} \times \mathcal{P}$.

1. If $R$ is a parameterized bisimulation, then, for any environment $e \in \text{Env}$, $R_e$ is a 0-parameterized bisimulation.
2. If $R$ is a parameterized bisimulation, $e \in \text{Env}$, $R_e$ is a $\lambda$-parameterized bisimulation, if and only if $R_e$ is $R_e^{-1}$.
3. If, for $e \in \text{Env}$, $R_e^i$ is a $\lambda_i$-parameterized bisimulation ($i = 1, 2$), then $R_e^1 \circ R_e^2$ is a $\lambda_1 + \lambda_2$-parameterized bisimulation. In particular, if $\rho$ is an ultrametric and $R_1$ and $R_2$ are all $\lambda$-bisimulation, so is $R_1 \circ R_2$.
4. If, for $e \in \text{Env}$, $R_e^i$ is a $\lambda$-bisimulation ($i \in I$), so is $\bigcup_{i \in I} R_e^i$.

Using the concept of $\lambda$-bisimulation, we can define the notion of $\lambda$-parameterized bisimulation in the usual way over the environment $e$.

**Definition 15.** Let $R$ be an Env-indexed family of binary relations on $\mathcal{P} \times \mathcal{P}$. For any $\lambda \in [0, \infty)$, we define $\lambda$-bisimulation over the environment $e$ as

$$\lambda_e = \bigcup \{R_e^i \mid R_e^i \text{ is a } \lambda_i \text{-parameterized bisimulation over the environment } e \}$$

(10)

In other words, if $(P, Q) \in R_e$, then $P$ and $Q$ are said to be $\lambda$-bisimilar over the environment $e$ whenever $\lambda$-parameterized bisimulation $R$ exists such that $(P, Q) \in R_e$. If $(P, Q) \in R_e$, $e$ can consume some actions to $e'$, but $P$ and $Q$ do not have any transition such that their next states are included in $R_{e'}$, then $P$ and $Q$ will never be considered when $P$ and $Q$ are executed in the environment $e$.

Next, we illustrate Definitions 9 and 15 with the following example.

**Example 16.** Two vending machines are assumed to exist. They can be expressed by the following process expressions:

$$P = 1d.\text{Coca-Cola}.\text{Collect}.P$$

$$Q = (1.2d.\text{Coca-Cola}.\text{Collect} + 0.7.\text{Fanta}.\text{Collect}) \cdot Q$$

(11)

A person who wants to buy a cup of Coca-Cola can be treated as an environment of the vending machines. According to CCS, the behaviors of the person can be described as a process:

$$e = 1d.\text{Coca-Cola}.\text{Collect}$$

(12)

The transition diagrams are described in Figure 4. Metric $\rho$ exists on the set of actions $\text{Act}$, where $\text{Act} = \{1d, 1.2d, 0.7d, \text{Coca-Cola}, \text{Fanta}, \text{Collect}\}$. Consider $\rho(1d, 1.2d) = 0.2, \rho(1d, 0.7) = 0.3, \rho(1.2d, 0.7d) = 0.5$, and $\rho(\text{Coca-Cola}, \text{Fanta}) = 0.2$. The distance between other actions is $\infty$.

The following relations can be defined:

$$R_{e_1} = \{(P_0, Q_0)\}$$

$$R_{e_1} = \{(P_1, Q_1), (P_1, Q_2)\}$$

$$R_{e_2} = \{(P_2, Q_1), (P_2, Q_3)\}$$

(13)

We can get that $b_{R_{e_0}} = 0.3, b_{R_{e_1}} = 0.2, b_{R_{e_2}} = 0$. So, $P_0 \sim_{e_0} Q_0, P_1 \sim_{e_1} Q_1, P_1 \sim_{e_1} Q_2, P_2 \sim_{e_2} Q_3, \text{ and } P_2 \sim_{e_2} Q_4$. The value means that when the person does not do anything, the approximate degree between two vending machines is 0.7. When the person puts 1d, then the approximate degree between two vending machines is 0.8. Finally, when the person chooses the Coca-Cola, the distance between them is 0.

Next, we will try to prove various properties of $\lambda$-parameterized bisimulation over environment $e$.

**Proposition 17.** Consider

1. If $e \sim_0 e'$. If $\lambda_1 \leq \lambda_2$, then $\sim_{e'} \subseteq \sim_{e}$. If $\lambda_1 \leq \lambda_2$, then $\sim_{e'} \subseteq \sim_{e}$.
2. For any $\lambda \in [0, \infty)$, $\sim_{e}^\lambda$ is a $\lambda$-parameterized bisimulation and it is reflexive and symmetric; $\sim_{e}^\lambda \circ \sim_{e}^\lambda = \sim_{e}^{\lambda + \lambda}$.

**Proposition 18.** Let $e \in \text{Env}$, $s_1 \sim_{e}^\lambda s_2$ if and only if, for any $\varepsilon \Rightarrow e'$, $b_{e' \rightarrow_{(\alpha \in \alpha', \omega)}}(s_1, s_2) \leq \lambda$ and $b_{e' \rightarrow_{(\alpha \in \alpha', \omega)}}(s_2, s_1) \leq \lambda$. 

\[1.2d \rightarrow 0.7d\]
Proof. If $s_1 \sim_1 s_2$, then we have that $b_{(s_1, s_2)}(s_1, s_2) \leq \lambda$ and $b_{(s_1, s_2)}(s_2, s_1) \leq \lambda$.

Conversely, we define that $(s_1, s_2) \in R_e$ if and only if $b_{(s_1, s_2)}(s_1, s_2) \leq \lambda$ and $b_{(s_1, s_2)}(s_2, s_1) \leq \lambda$ for all $e \Rightarrow e'$. From $b_{(s_1, s_2)}(s_1, s_2) \leq \lambda$, we know that, for any $s_1' \in S$ and $t \in \text{Act}$ with $s_1' \xrightarrow{t} s_1$, if $t = \alpha$, then, for any $n \geq 1$, there exist $s_2' \in S$ and $t \in \text{Act}$ such that $s_2' \xrightarrow{t} s_2'$ and $\rho(t, t') < \lambda + 1/n$ and $s_1' \sim_1 s_2'$. If $t \neq \alpha$, then, for any $n \geq 1$, there exist $s_2' \in S$ and $t \in \text{Act}$ such that $s_2' \xrightarrow{t} s_2'$ and $\rho(t, \alpha) \leq \lambda + 1/n$, and $s_1' \sim_1 s_2'$. By noting that $s_1' \sim_1 s_2'$ implies $s_1'R_1s_2'$, we obtain $b_{(s_1, s_2)}(s_1, s_2) \leq \lambda$. Similarly, $b_{(s_1, s_2)}(s_2, s_1) \leq \lambda$. Therefore, $R_e$ is a $\lambda$-parameterized bisimulation over the environment $e$, and $s_1 \sim_1 s_2$.

4. Congruence of $\lambda$-Parameterized Bisimulation over Environment $e$

In order to support hierarchic development and modular decomposition of software, it is necessary to ensure that equivalences are congruent with respect to processes combinators. It means that if two processes are equivalent, then the new processes obtained by combining the given processes are also equivalent. In this section, we will mainly discuss these substitutivity laws of $\lambda$-parameterized bisimulation under various combinators.

Definition 19 ($\lambda$-round [7]). Let $(M, \rho)$ be a metric space; $Y \subseteq M, \lambda \geq 0$. If, for any $x, y \in M, x \in Y$ and $\rho(x, y) \leq \lambda$ implies $y \in Y$, then $Y$ is said to be $\lambda$-round. If, for some $\mu > \lambda, Y$ is $\mu$-round, then $Y$ is said to be strongly $\lambda$-round.

Definition 20 (isomorphism mapping [7]). Let $(M, \rho)$ be a metric space. $f$ is a mapping from $M$ into itself. If, for any $x, y \in M, \rho(f(x), f(y)) = \rho(x, y)$, then $f$ is said to be isomorphism mapping.

From the definition above, we can see that $\lambda$-round is a rigorous condition. In [7], they prove that there are only two $\lambda$-round sets in the real line when $\lambda > 0$. One is the empty set and the other is the real line itself. They also show that it is not the same case as in the real line in general. For example, if $M = \bigcup_{i \in I} M_i$ and $[M_i]_{i \in I}$ is pairwise disjoint and $\rho(x, y) > \mu$ for any $x, y \in M$ with $x \in M_i$ and $y \in M_j (i \neq j)$, then each $M_i$ is $\lambda$-round for every $\mu < \lambda$.

We now consider the substitutivity laws of $\lambda$-parameterized bisimulation under various combinators in our process calculi.

Proposition 21. Let $e = (\text{Env}, \text{Act}, \Rightarrow)$ be an environment transition system; $e \in \text{Env}$,

1. If $P_1 \sim_1 P_2$ and $e' \Rightarrow e$, then $\alpha \cdot P_1 \sim_1 \alpha \cdot P_2$.
2. If $L$ is $\lambda$-round and $P_1 \sim_1 P_2$, then $P_1 \parallel L \sim_1 P_2 \parallel L$.
3. If $f$ is isomorphism mapping on Act and $P_1 \sim_1 \lambda^{-1}(f)P_2$, then $P_1[f] \sim_1 \lambda^{-1}P_2[f]$.
4. If $P_1 \sim_1 P_2$, then $W + P_1 \sim_1 W + P_2$.

Proof. (1) We need to show $b_{(e')_{(s_1, s_2)}}(\alpha \cdot P_1, \alpha \cdot P_2) \leq \lambda$ and $b_{(e')_{(s_1, s_2)}}(\alpha \cdot P_2, \alpha \cdot P_1) \leq \lambda$. By Definition 2, $\alpha \cdot P_1 \Rightarrow P_1$ and $P_2 \Rightarrow P_2$, so there exists $P_2$ such that $\alpha \cdot P_2 \Rightarrow P_2$ with $\rho(\alpha, \alpha) \leq \rho(\alpha, \alpha)$ and $\rho(\alpha, \alpha) = 0 \leq 0$. So $b_{(e')_{(s_1, s_2)}}(\alpha \cdot P_1, \alpha \cdot P_2) = 0 \leq \lambda$. Similarly, $b_{(e')_{(s_1, s_2)}}(\alpha \cdot P_2, \alpha \cdot P_1) = 0 \leq \lambda$.

(2) Let $e \in \text{Env}$, $S_e = \{P_1 \parallel L, P_2 \parallel L : P_1 \sim_1 P_2\}$. Next, we need to show $b_{S_e} \leq \lambda$.

Let $f \Rightarrow f$. If $P_1 \parallel L \Rightarrow P_1'$, then $\alpha, \alpha' \not\Rightarrow L \cup L$, $P_1 \Rightarrow P_1'$, and $P_1' \Rightarrow P_1''$. Since $L$ is $\lambda$-round, $u, u \not\Rightarrow L \cup L$. By the transition rule, $P_1' \parallel L \Rightarrow P_1'' \parallel L = P_1''$ with $\rho(u, \beta) \leq \rho(\alpha, \beta)$, $\rho(\alpha, \beta) \leq \lambda$, and $P_1'' \sim_1 P_2''$. Since $L$ is $\lambda$-round, $u, u \not\Rightarrow L \cup L$. By the transition rule, $P_1 \parallel L \Rightarrow P_1'' \parallel L = P_1''$. Therefore, $b_{S_e} \leq \lambda$.

(3) For any $e \in \text{Env}$, let $S_e = \{P_1[f], P_2[f] : P_1 \sim_1 \lambda^{-1}(f)P_2\}$. Next, only need to show $b_{S_e} \leq \lambda$.

In fact, let $e \Rightarrow f$. If $P_1[f] \Rightarrow P_1'$, then there exists $h \in \text{Act}$, such that $P_1 \Rightarrow P_1'$, and $f(\alpha) = f(h)$. If $\alpha \neq b$, since $P_1 \Rightarrow P_1'$, $f^{-1}(b) \Rightarrow f^{-1}(b)$ and there exist $P_1' \in S$, and $u \in \text{Act}$ such that $P_1' \parallel L \Rightarrow P_1'' \parallel L = P_1''$ with $\rho(u, \beta) \leq \rho(\alpha, \beta)$, $\rho(\alpha, \beta) \leq \lambda$, and $P_1'' \sim_1 P_2''$. Since $L$ is $\lambda$-round, $u, u \not\Rightarrow L \cup L$. By the transition rule, $P_1'[f] \Rightarrow P_1''[f]$. Let $P_1'' [f] \Rightarrow P_1''[f]$ and $f$ is isomorphism, $\rho(f(u), b) = \rho(f(f(u)), f^{-1}(b)) = \rho(u, f^{-1}(b)) \leq \rho(u, h^{-1}(b)) = \rho(f(h), h) = \rho(f(u), h) = \rho(u, h) \leq \lambda$, and $(P_1''[f], P_1''[f]) \in S_g$.

On the other hand, if $\alpha = b$, $P_1 \sim_1 \lambda^{-1}(f)P_2$ tells us $f^{-1}(e) \Rightarrow f^{-1}(e)$ and there exist $P_1' \in S$, and $u \in \text{Act}$ such that $P_1 \Rightarrow P_1'$, with $\rho(u, h) \leq \lambda$, $(P_1''[f], P_1''[f]) \in S_g$. Therefore, $b_{S_e} \leq \lambda$.
On the other hand, if \( \alpha = b \), then \( P_1 \overset{\alpha}{\rightarrow} P'_1 \) leads to the idea that there exist \( P'_2 \in \mathcal{P} \) and \( \beta \in \text{Act} \) such that \( P_2 \overset{\beta}{\rightarrow} P'_2 \), with \( \rho(\alpha, \beta) \leq \lambda \) and \( (P'_1, P'_2) \in \sim_f \). So, \( W + P_2 \overset{\beta}{\rightarrow} W \), and \( b_{(\sim_f(\alpha, \beta))} (W + P_1, W + P_2) \leq \lambda \). And \( W \overset{\alpha}{\rightarrow} P' \) tells us \( W + P_2 \overset{\alpha}{\rightarrow} (W + P_1, W + P_2) \leq \lambda \).

Similarly, we can obtain that \( b_{(\sim_f(\alpha, \beta))} (W + P_2, W + P_1) \leq \lambda \). So, \( W + P_2 \overset{\alpha}{\rightarrow} W + P_2 \).

5. Parameterized Probabilistic Bisimulation

The behaviors of some complicated software systems are often probabilistic in nature. Usually, a system with probabilistic behaviors may typically be described as a probabilistic process. van Glabbeek et al. [31] introduced three models of probabilistic processes in accordance with the relation between the occurrences of actions and transition probabilities: a probabilistic processes in accordance with the relation between the probabilistic processes are reviewed. As in SCCS [33], let \( \rho(\alpha, \beta) \leq \lambda \) for \( \alpha \in \text{Act} \), denote products of actions, for example, \( \alpha \beta \).

An expression that has no free variables is called a process, and \( \text{Pr} \) is the set of all PCCS processes. Intuitively, \( \text{Pr} \) has no derivations, whereas \( \alpha : E \) performs action \( \alpha \) with probability \( 1 \) and then behaves like \( E \). A summation expression offers a probabilistic choice among its constituent behaviors, where \( I \) is accountable, so \( \sum \in \text{Pr} \) is a probabilistic distribution. When \( I = 0 \), then \( 0 = \sum \in \text{Pr} \).

Let \( \text{DP} \) be the class of deterministic PCCS processes; that is, if \( P \in \text{DP} \), then, for any \( \alpha \in \text{Act} \), \( P \) has at most one probabilistic derivation of type \( \alpha \). Then, the operational semantics of deterministic PCCS process can be described as follows.

**Definition 23** (see [13]). The structure operational semantics of deterministic PCCS process based on probabilistic derivation is given as a set of inference rules, in the style of Plotkin:

\[
\begin{align*}
\text{Act} & : \alpha \cdot E \overset{\alpha[1]}{\rightarrow} E, \\
\text{Sum} & : E_j \overset{[\alpha]}{\rightarrow} E' \implies \sum_{i \in I} [p_i] E_i \overset{[\alpha[p]]}{\rightarrow} E'( j \in I), \\
\text{Pro} & : E \overset{\alpha[p]}{\rightarrow} E', \\
F \overset{\beta[q]}{\rightarrow} F' & \implies F \times F \overset{\alpha[p, \beta[q]]}{\rightarrow} E' \times F', \\
\text{Res} & : E \overset{[\alpha]}{\rightarrow} E' \implies E \upharpoonright A \overset{[\alpha[p \upharpoonright A]]}{\rightarrow} E' \upharpoonright A, \text{ where} \\
\nu (E, A) & = \sum \left\{ [p_i] \mid E \overset{[\alpha[p]]}{\rightarrow} E', \alpha \in A \right\}, \\
\text{Rel} & : E \overset{[\alpha]}{\rightarrow} E' \implies E \left[ f \right] \overset{f(\alpha)[p]}{\rightarrow} E' \left[ f \right], \\
\text{Con} & : \sum_{X \in \text{Act}} \left\{ \text{fix}_X E \right\} \overset{[\alpha]}{\rightarrow} E' \implies \text{fix}_X E \overset{[\alpha]}{\rightarrow} E'.
\end{align*}
\]

Similar to the assumption on parameterized bisimulation, the assumption that an environment and its behaviors can be described as a deterministic PCCS process is reasonable. Env is the set of environments, Act is the set of actions (identical to the set of actions used in the transition system of processes),
and ⇒ is a subset of Env × Act × [0, 1] × Env, and e ⇒ e’ means that e may consume the action α with the probability p and after that becomes the environment e’.

Then, parameterized probabilistic bisimulation is defined.

**Definition 24.** Let ε = (Env, Act, ⇒) be a probabilistic environment transition system. Then an ε-parameterized probabilistic bisimulation, R, is an Env-indexed family of binary relations R_e ⊆ DPr × DPr for e ∈ Env, such that

\[\mathcal{R}_e^{\text{pr}} = \{R_e | R_e \text{ is parameterized probabilistic bisimulation over the environment } e\}.\]  

Two processes, P and Q, are said to be probabilistic bisimulation equivalence on the environment e ∈ Env if and only if there exists ε-parameterized probabilistic bisimulation R such that (P, Q) ∈ R_e. We write parameterized probabilistic bisimulation by using the following signal, P ∼_{e}^{\text{pr}} Q, to distinguish the difference between parameterized bisimulation and parameterized probabilistic bisimulation.

**Example 25.** Let e, P, and Q be given in Figure 5. The Env-indexed family R is shown as follows:

\[\begin{align*}
R_{e_0} &= \{(P_0, Q_0)\} \\
R_{e_1} &= \{(P_1, Q_1), (P_2, Q_2), (P_1, Q_2)\} \\
R_{e_2} &= \{(P_2, Q_2)\} \\
R_{e_3} &= \{(P_3, Q_3), (P_2, Q_3)\} \\
R_{e_4} &= \{(P_4, Q_4), (P_2, Q_4)\}.
\end{align*}\]

Then, according to Definition 24, R is ε-parameterized probabilistic bisimulation.

Next, we try to relax the ε-parameterized probabilistic bisimulation to establish the approximate parameterized probabilistic bisimulation. In [18], there is a rather rich state-space structure and a metric between two processes employs the probabilistic conditional kernels underlying the two stochastic processes. Another metric is based on the dynamical properties of the two processes. Compared with the metric in [18], our model only focuses on the difference of probabilities where two processes can execute the same action.

**Definition 26.** Let ε = (Env, Act, ⇒) be a probabilistic environment transition system. Consider λ ∈ [0, 1]. R is an Env-indexed family of binary relations, R_e ⊆ DPr × DPr. If, for any e ∈ Env, e ⇒ e’ and (P, Q) ∈ R_e, the following conditions are satisfied:

1. If P \xrightarrow{\alpha[p]} P’, then there exists Q’ ∈ DPr such that Q \xrightarrow{\alpha[q]} Q’, max||p − r|, |q − r|, |p − q|| ≤ λ, and (P’, Q’) ∈ R_e.

2. If Q \xrightarrow{\alpha[q]} Q’, then there exists P’ ∈ DPr such that P \xrightarrow{\alpha[p]} P’, max||p − r|, |q − r|, |p − q|| ≤ λ, and (P’, Q’) ∈ R_e.

Then we call R_e λ-probabilistic bisimulation. P and Q are said to be λ-probabilistic bisimilar on the environment e if λ-probabilistic bisimulation R_e exists such that (P, Q) ∈ R_e.

**Proposition 27.** Let ε = (Env, Act, ⇒) be a probabilistic environment transition system. Consider λ ∈ [0, 1]. R is an Env-indexed family of binary relations; R_e ⊆ DPr × DPr:

1. If R is a parameterized probabilistic bisimulation, then, for e ∈ Env, R_e is 0-probabilistic bisimulation.
(2) For $e \in \text{Env}$, $R_e$ is $\lambda$-probabilistic bisimulation if and only if so is $R_e^{-1 \cdot \lambda}$.

(3) For $e \in \text{Env}$, $R_e^1$ is a $\lambda_1$-probabilistic bisimulation ($i = 1, 2$); then $R_e^1 \circ R_e^2$ is a max$[\lambda_1, \lambda_2]$-probabilistic bisimulation.

(4) For $e \in \text{Env}$, $R_e^i (i \in I$, where $I$ is indexed set) is $\lambda$-probabilistic bisimulation and so is $\bigcup_{e \in I} R_e^i$.

Definition 28. Let $e = (\text{Env}, \text{Act}, \Rightarrow)$ be a probabilistic environment transition system. Consider $\lambda \in [0, 1]$. $R$ is an $\text{Env}$-indexed family of binary relations, $R_e \subseteq \text{DPr} \times \text{DPr}$; we define

$$\sim_e = \{ R_e \mid R_e \text{ is } \lambda\text{-probabilistic bisimulation} \}.$$ (18)

If $P$ and $Q$ are said to be $\lambda$-probabilistic bisimilar on the environment $e$, then $\lambda$-probabilistic bisimulation $R_e$ exists such that $(P, Q) \in R_e$. Thus, we can write $P \sim_{\sim_e} Q$.

Proposition 29. Consider

(1) $\sim_e$ forms a sequence of successively larger relations; that is, $\lambda \leq \lambda'$ implies $\sim_{\lambda'} \subseteq \sim_{\lambda'}$.

(2) $\sim_{\lambda} = \bigcap_{\lambda > 0} \sim_{\sim_{\lambda}} \supseteq \sim_{\lambda}^0$ coincides with the parameterized probabilistic bisimulation.

Example 30. Let $e$, $P$, and $Q$ be given in Figure 6. The $\text{Env}$-indexed family $R$ is shown as follows:

$$R_0 = \{ (P_0, Q_0) \},$$

$$R_1 = \{ (P_1, Q_1), (P_2, Q_2), (P_3, Q_3) \},$$

$$R_2 = \{ (P_2, Q_2) \},$$

$$R_3 = \{ (P_3, Q_3), (P_3, Q_3), (P_4, Q_4) \}.$$ (19)

Then we can obtain that $P_{e,0} \sim_{\sim_{e,0}} Q_0$, $P_{e,1} \sim_{\sim_{e,1}} Q_1$, $P_{e,2} \sim_{\sim_{e,2}} Q_2$, but $P_{e,3} \sim_{\sim_{e,3}} Q_2$.

Theorem 31. Let $e = (\text{Env}, \text{Act}, \Rightarrow)$ be a probabilistic environment transition system; $\lambda \in [0, 1]$:

(1) If $P \sim_{\sim_{e,0}} Q$, then $\alpha \cdot P \sim_{\sim_{e,0}} \alpha \cdot Q$.

(2) If $f$ is isomorphism mapping on Act and $P \sim_{\sim_{e,0}} Q$, then $P[f] \sim_{\sim_{e,0}} Q[f]$.

(3) If $P \sim_{\sim_{e,0}} Q$ and $P \sim_{\sim_{e,0}} T$, then $P + (1 - p) R_{\sim_{e,0}} (1 - p) T$.

Proof. (1) It only needs to show there is Env-parameterized probabilistic bisimulation $R; R_{\sim_{e,0}}$ is $\lambda$-probabilistic bisimulation such that $(P \cdot e, Q \cdot e) \in R_{\sim_{e,0}}$. Since $P \sim_{\sim_{e,0}} Q$, Env-parameterized probabilistic bisimulation $S$ exists such that $(P, Q) \in S$. Then, we construct Env-parameterized probabilistic bisimulation $R$ which is defined as $R_{\sim_{e,0}} = \{ (P \cdot e, Q \cdot e) \mid (P, Q) \in S \}$, for any $e \in \text{Env}$, $R_e = S$. Thus, we need to prove that $R_{\sim_{e,0}}$ is $\lambda$-probabilistic bisimulation. In fact, according to Definition 23, we have that $R_{\sim_{e,0}}$ is isomorphism mapping on $\text{Act}$, such that $(P \cdot e, Q \cdot e) \in R_{\sim_{e,0}}$. So, $R_{\sim_{e,0}}$ is $\lambda$-probabilistic bisimulation.

(2) Env-parameterized probabilistic bisimulation $R$ must be presented, and $R_{f(e)}$ is $\lambda$-probabilistic bisimulation such that $(P[f], Q[f]) \in R_{f(e)}$. In fact, since $P \sim_{\sim_{e,0}} Q$, Env-parameterized probabilistic bisimulation $S$ exists and $S_e$ is $\lambda$-probabilistic bisimulation such that $(P, Q) \in S_e$. Then, we establish Env-parameterized probabilistic bisimulation $R$ defined as $R_{f(e)} = \{ (P[f], Q[f]) \mid (P, Q) \in S_e \}$. $R_{f(e)} = S_e$. Then, $R_{f(e)}$ must be proved to be $\lambda$-probabilistic bisimulation.

In fact, if $f(e) \Rightarrow (e')$, then $e' \Rightarrow f'(\beta)'$ and $\alpha = f(\beta)$. If $P[f] \Rightarrow \beta[\alpha]$ $P'$, then $P[\beta[\alpha]] \Rightarrow [\\beta[\alpha]], P' \Rightarrow [\\beta[\alpha]], P' \Rightarrow [\\beta[\alpha]], P' \Rightarrow [\\beta[\alpha]]$. Since $(P, Q) \in S$, $Q'[\alpha]\Rightarrow [\\beta[\alpha]]$, $P'[\alpha]\Rightarrow [\\beta[\alpha]]$, $P'[\alpha]\Rightarrow [\\beta[\alpha]]$. Furthermore, according to Definition 23, we can obtain that $Q[f] \Rightarrow [\\beta[\alpha]]$, $Q'[\alpha]\Rightarrow [\\beta[\alpha]]$, and $(P'[\alpha], Q'[\alpha]) \in R_{f(e)}$. Thus, $R_{f(e)}$ is $\lambda$-probabilistic bisimulation, so $P[f] \sim_{\sim_{e,0}} Q[f]$.

(3) We need to determine Env-parameterized probabilistic bisimulation $R; R_{f(e)}$ is $\lambda$-probabilistic bisimulation such that $(P + (1 - p) R, p + (1 - p) T) \in R_{f(e)}$. In fact, since $P \sim_{\sim_{e,0}} Q$, $R_{\sim_{e,0}} T$, so there are Env-parameterized $S_1$ and $S_2$ such that $(P, Q) \in S_1 \subseteq S_1$, $(R, T) \in S_2 \subseteq S_2$. Next, Env-parameterized probabilistic bisimulation $R$ is defined as $R_{f(e)} = \{ (P + (1 - p) R, p Q + (1 - p) T) \mid (P, Q) \in S_1 \}$. $R_e = S_2$, and $R_e = S_2$. 

\[ \text{FIGURE 6: } P_0 \text{ and } Q_0 \text{ are } 0.3\text{-probabilistic bisimulation on the environment } e_0. \]
If $pe + (1 - p)f \Rightarrow e'$, then, according to Definition 23, two cases exist.

Case 1 ($e \Rightarrow e'$ and $r = pm$). If $pP + (1 - p)R \Rightarrow [f]$ then $P \Rightarrow [f]$ or $R \Rightarrow [f]$ and $h = pl$ or $h = (1 - p)v$.

When $P \Rightarrow [f]$, since $(P, Q) \in S^1$, so there is $Q \in DPr$ such that $Q \Rightarrow [f']$ and $\max{|l - m|, |k - m|, |k - l|} \leq \lambda$, and $(P', Q') \in S^2$.

So, according to Definition 23, we can get $pQ + (1 - p)R \Rightarrow [f']$ and $\max{|h - r|, |pk - r|, |h - pk|} = p \max{|l - m|, |k - m|, |l - k|} \leq \lambda$ and $(P', Q') \in S^2$.

Case 2 (there is $f \Rightarrow f'$ and $r = (1 - p)m$). It is similar to Case 1.

Notice that we do not show the proof of the substitutivity law of the synchrony operator $\times$ and restrict operator $\mid$. The main reason is that we can not find a suitable $\lambda, \epsilon$-parameterized probabilistic bisimulation according to Definition 24. Therefore, our model only uses prefix operator $\alpha$, relabeled operator $\lceil f \rceil$, and sum operator $\Sigma$ to help us verify the approximate correctness. In the future, I will attempt to find a suitable model to support the substitutivity law.

6. Conclusion

In this study, we formalized the approximate correctness of software products. We focused on the approximate version of parameterized bisimulation and extended parameterized bisimulation to probabilistic settings. We presented the definitions of $\lambda$-parameterized bisimulation and $\lambda$-parameterized probabilistic bisimulation over the environment $e$. These approximate versions provide theoretical foundations to verify the degree to which the software is approximately correct. In the future, we will try to find some effective algorithms to realize the verification. The substitutivity laws of $\lambda$-parameterized bisimulation over the environment $e$ and $\lambda$-parameterized probabilistic bisimulation over the environment $e$ are presented.

The modal logical characterizations of $\lambda$-parameterized bisimulation and $\lambda$-parameterized probabilistic bisimulation over the environment $e$ are useful to verify whether or not two processes are approximate under parameterized bisimulation. In the future, we will attempt to establish the modal logical descriptions of $\lambda$-parameterized bisimulation and $\lambda$-parameterized probabilistic bisimulation.

Conflict of Interests

The author declares that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The work is supported by the NSFC (61300048, 61021004), the Anhui Provincial Natural Science Foundation (1308085QF117, 1508085MA14), the Key Natural Science Foundation of Universities of Anhui Province (KJ2014A223, KJ2011A248), the Excellent Young Talents in Universities of Anhui Province, and the Major Teaching Reform Project of Anhui Higher Education Revitalization Plan (2014ZDY058).

References


