Characterization and Verification of Stuttering Equivalence

Xinxin Liu(✉) and Wenhui Zhang

State Key Laboratory of Computer Science, Institute of Software,
Chinese Academy of Sciences , University of Chinese Academy of Sciences,
Huairou, China
{xinxin,zwh}@ios.ac.cn

Abstract. Stuttering equivalence is an important equivalence relation on Kripke structures. It is the equivalence which preserves all CTL*-X properties. Two key issues concerning this equivalence are how to characterize it and how to verify whether two given states are equivalent with respect to it. For this purpose, we propose two bisimulation style definitions, one called ω-bisimulation which provides a concise characterization of the equality and one called stuttering bisimulation with induction which provides a verification method for establishing the equality. We also show that stuttering bisimulation with induction coincides with well-founded bisimulation, a notion introduced by Namjoshi for verifying stuttering equivalence.

1 Introduction

Stuttering equivalence on Kripke structures is an important equivalence that has the exact distinguishing strength of the set of CTL*-X properties (the computation tree logic CTL* [14] without the next operator). An important feature of stuttering equivalence is that it is a divergence preserving equivalence with a high level of abstraction. A main issue concerning stuttering equivalence, which has both theoretical interest and practical implication, is how to characterize such an equivalence in a way that two states can be shown equivalent with minimal effort. The definition of stuttering equivalence in [3] presented it as the limit of a converging sequence of equivalences. Such a definition is not very helpful either for equality checking or for equality proving. Here "equality checking" is the problem of deciding whether two given states are equivalent, and "equality proving" is the problem of verifying that whether a given evidence (or a proof, e.g. a supposed bisimulation relation) of equality two states is valid. In [11], a simpler characterization is proposed, in which a well founded relation is used for the characterization of the stuttering equivalence. It is called well-founded bisimulation. It is proven that well-founded bisimulation corresponds to stuttering equivalence [11]. A difficulty with this definition is that it is not obvious how to construct the well-founded relation, in order to be used to show the equivalence of states. In addition, such a well-founded relation may be unnecessarily large, if it is not

C. Jones et al. (Eds.): Zhou-Festschrift, LNCS 11180, pp. 116–132, 2018.
https://doi.org/10.1007/978-3-030-01461-2_7

constructed carefully. In this paper, we address these difficulties by proposing a characterization of stuttering equivalence with the notion of stuttering bisimulation with induction. Firstly, we propose another characterization with the notion of ω-bisimulation which is easy to use to establish theoretical foundation. Secondly, we have a concept that uses the inductive principle instead of infinite sequence of relations and well-founded relations. Thirdly, we relate well-founded bisimulation to the new definition. Finally, we show that the new definition can be used to construct a well-founded relation that is required for showing stuttering equivalence by well-founded bisimulation, and therefore providing a method for proving the equivalence of states with well-founded bisimulation.

The paper is organized as follows. In the next section we present stuttering equivalence in terms of ω-bisimulation. In Sect. 3 we introduce the notion of stuttering bisimulation with induction to characterize stuttering equivalence. In Sect. 4 we study the relationship between stuttering bisimulation with induction and well-founded bisimulation. We discuss related works in Sect. 5, summarize and conclude in Sect. 6.

2 Stuttering Equivalence and ω-Bisimulation

In this section we establish a theoretical foundation for stuttering equivalence with the notion of ω-bisimulation. We start with some basic notions and notations.

Definition 1 *(*Kripke structure and infinite runs*). A* Kripke structure *is a tuple* $\mathcal{K} = \langle S, AP, L, \longrightarrow \rangle$ *where:*

- S *is a set of* states*;*
- AP *is a set of* atomic propositions *or* labels*;*
- $L : S \to 2^{AP}$ *is the labeling function which assigns each state a set of atomic propositions;*
- $\longrightarrow \subseteq S \times S$ *is the* transition relation*. An element* (s, t) *of* $\longrightarrow$*, usually written as* $s \longrightarrow t$*, is called a* transition*. Following the convention, we assume that* $\longrightarrow$ *is a total relation, i.e. for each* $s \in S$ *there exists* $s' \in S$ *such that* $s \longrightarrow s'$*.*
- *A finite run of* $\mathcal{K}$ *is a finite sequence of states, with each pair of neighbouring states connected by a transition. If* ρ *is a finite run of* $\mathcal{K}$ *with starting state* s *and finishing state* t*, we also say that* ρ *is a finite run from* s *to* t*, and write* $first(\rho)$ *for* s*,* $last(\rho)$ *for* t*. The length of* ρ*, written* $length(\rho)$*, is the number of transitions connecting* ρ *(thus* $length(s) = 0$ *where* s *is the run consists of a single state* $s \in S$*).*
- *An infinite run of* $\mathcal{K}$ *is an infinite sequence of states, with each pair of neighbouring states connected by a transition. If* ρ *is an infinite run of* $\mathcal{K}$ *with starting state* s*, we also say that* ρ *is an infinite run from* s*, and write* $first(\rho)$ *for* s*.*

Let R, R_1, R_2 be binary relations. In general, we define the *converse* R^{-1} of R and the *composition* R_1R_2 of R_1 and R_2 by

$$\begin{aligned} R^{-1} &= \{(s,t) \mid (t,s) \in R\}, \\ R_1R_2 &= \{(s,t) \mid \text{there exist } (s,u) \in R_1, (u,t) \in R_2 \text{ for some } u\}. \end{aligned}$$

Definition 2 *(*Relations on finite and infinite runs*). Let $\mathcal{K} = \langle S, AP, L, \longrightarrow \rangle$ be a Kripke structure, R be a binary relation on S.*

Define a binary relation $R^\natural$ between finite runs of $\mathcal{K}$ by induction on the lengths of the runs such that $(s_1 \dots s_n, t_1 \dots t_m) \in R^\natural$ if and only if $(s_1, t_1) \in R$ and moreover one of the following holds:

1. $n = m = 1$;
2. $(s_2 \dots s_n, t_1 \dots t_m) \in R^\natural$;
3. $(s_1 \dots s_n, t_2 \dots t_m) \in R^\natural$;
4. $(s_2 \dots s_n, t_2 \dots t_m) \in R^\natural$.

Define a binary relation $R^\sharp$ between infinite runs of $\mathcal{K}$ such that for two infinite runs σ, ρ of $\mathcal{K}$, $(\sigma, \rho) \in R^\sharp$ if and only if both of the following hold:

1. *for each finite prefix σ' of σ, there is a finite prefix ρ' of ρ with $(\sigma', \rho') \in R^\natural$;*
2. *for each finite prefix ρ' of ρ, there is a finite prefix σ' of σ with $(\sigma', \rho') \in R^\natural$.*

The intuition of $R^\natural$ ($R^\sharp$) is that it describes a relation between two finite (infinite) runs such that the states in the two sequences of the runs are pairwise related by R while making progress in lock steps modulo finite stuttering.

The following lemma shows expected homomorphic and monomorphic properties of $\natural$ and $\sharp$, which is required later in establishing transitivity of some desired equivalence relation.

Lemma 1. *Let $\mathcal{K} = \langle S, AP, L, \longrightarrow \rangle$ be a Kripke structure, R_1, R_2 be two binary relations on S. Then*

1. ${R_1}^\natural {R_2}^\natural \subseteq (R_1R_2)^\natural$;
2. ${R_1}^\sharp {R_2}^\sharp \subseteq (R_1R_2)^\sharp$;
3. *If $R_1 \subseteq R_2$, then ${R_1}^\natural \subseteq {R_2}^\natural$ and ${R_1}^\sharp \subseteq {R_2}^\sharp$.*

Proof. 1 can be proved by a detailed case analysis. 2 follows from 1. For 3, suppose $R_1 \subseteq R_2$, then it can be proved by induction on the total lengths of the finite runs σ and ρ that if $(\sigma, \rho) \in R_1^\natural$ then $(\sigma, \rho) \in R_2^\natural$. Then it follows immediately that in this case also ${R_1}^\sharp \subseteq {R_2}^\sharp$. □

In [3], stuttering equivalence was originally defined as the limit of a converging sequence of equivalences. This kind of definition is not easy to work with either for theoretical foundation or for practical verification. The notion of bisimulation, proposed by Park [7], has been very successfully applied by Milner in studying equivalence relations on labeled transition systems [8]. Thus we wish to establish the theory of stuttering equivalence based on the notion of bisimulation. However, due to the consideration of infinite runs in stuttering

equivalence, the bisimulation characterization of stuttering equivalence is a little more complex than that of many well-known equivalence relations. So we first ignore the issue of divergence, and present a divergence blind equivalence which is easy to describe by using bisimulation and which is also very close to stuttering equivalence.

Definition 3 *(*Stuttering bisimulation, divergence blind stuttering equivalence*). Let $\mathcal{K} = \langle S, AP, L, \longrightarrow \rangle$ be a Kripke structure. A* stuttering bisimulation *is a binary relation $R \subseteq S \times S$ such that for all $(s_0, t_0) \in R$ the following hold:*

1. *$L(s) = L(t)$;*
2. *if $s_0 \longrightarrow s_1$ then there is a finite run ρ from t_0 such that $(s_0 s_1, \rho) \in R^\natural$;*
3. *if $t_0 \longrightarrow t_1$ then there is a finite run σ from s_0 such that $(\sigma, t_0 t_1) \in R^\natural$.*

We write $s \approx_{db} t$ if there is a stuttering bisimulation R such that $(s, t) \in R$. We call $\approx_{db}$ divergence blind stuttering equivalence.

With this definition, by Lemma 1, it is routine to prove that $\approx_{db}$ is an equivalence relation, and $\approx_{db}$ is the largest stuttering bisimulation.

Definition 4 *(Divergence). Let $\mathcal{K} = \langle S, AP, L, \longrightarrow \rangle$ be a Kripke structure, $\equiv$ be an equivalence relation on S. An infinite run $\rho = s_0 s_1 \ldots s_i \ldots$ of $\mathcal{K}$ is called a* divergent run with respect to $\equiv$ *if $s_0 \equiv s_i$ for all i. In this case s_0 is said to be* divergent with respect to $\equiv$*, written $s_0 \Uparrow_\equiv$. We say that $\equiv$ is divergence preserving, if whenever $s \equiv t$ and $s \Uparrow_\equiv$ then $t \Uparrow_\equiv$.*

In discussing divergence we often omit mentioning "with respect to $\equiv$" if it is obvious from the context.

Although $\approx_{db}$ is an equivalence relation with many desired properties, it is well known that $\approx_{db}$ is not divergence preserving. This is why we call it *divergence blind* stuttering equivalence, and use the subscript *db* for it. In order to obtain divergence preserving property, it turns out that we need to strengthen the definition of divergence blind stuttering equivalence so that correspondence of all infinite runs from states are required, instead of only requiring correspondence of finite runs. The result is the following notion of ω-bisimulation.

Definition 5 *(ω-*bisimulation, stuttering equivalence*). Let $\mathcal{K} = \langle S, AP, L, \longrightarrow \rangle$ be a Kripke structure. An* ω-bisimulation *is a binary relation $R \subseteq S \times S$ such that for all $(s, t) \in R$ the following hold:*

1. *$L(s) = L(t)$;*
2. *for any infinite run σ from s, there exists an infinite run ρ from t such that $(\sigma, \rho) \in R^\sharp$;*
3. *for any infinite run ρ from t, there exists an infinite run σ from s such that $(\sigma, \rho) \in R^\sharp$.*

Define $\approx_{st} = \bigcup\{R \mid R$ is an ω-bisimulation$\}$. We call $\approx_{st}$ stuttering equivalence*, and for $s, t \in S$ we say s is* stuttering equivalent to t *if $s \approx_{st} t$.*

The name ω-bisimulation refers to the examination of infinite runs in the definition.

Theorem 1. *$\approx_{st}$ is an equivalence relation.*

Proof. First note that $Id = \{(s, s) \mid s \in S\}$ is an ω-bisimulation. Thus $\approx_{st}$ is reflexive.

If R is an ω-bisimulation then it is easy to see from the definition that its converse R^{-1} is also an ω-bisimulation. Thus $\approx_{st}$ is symmetric.

If R_1, R_2 are two ω-bisimulations, then by Lemma 1 (2) it is easy to see that their composition R_1R_2 is also an ω-bisimulation. Thus $\approx_{st}$ is transitive. □

Theorem 2. *Let $\mathcal{K} = \langle S, AP, L, \longrightarrow\rangle$ be a Kripke structure. Then $\approx_{st}$ is the largest ω-bisimulation on S.*

Proof: First, we show that $\approx_{st}$ is an ω-bisimulation. For that, let $s \approx_{st} t$ for $s, t \in S$. Then there is $R \subseteq S \times S$ such that $(s, t) \in R$ and R is an ω-bisimulation. By Definition 5 the following hold:

1. $L(s) = L(t)$;
2. for any infinite run σ from s, there exists an infinite run ρ from t such that $(\sigma, \rho) \in R^\sharp$;
3. for any infinite run ρ from t, there exists an infinite run σ from s such that $(\sigma, \rho) \in R^\sharp$.

Note that $R \subseteq \approx_{st}$, and by Lemma 1 (3) $R^\sharp \subseteq \approx_{st}^\sharp$, then it is easy to see that $\approx_{st}$ is an ω-bisimulation.

It is obvious from the definition that if R is an ω-bisimulation then $R \subseteq \approx_{st}$, thus $\approx_{st}$ is the largest such. □

This theorem shows that $\approx_{st}$ is well defined. Now we examine divergence preserving property of $\approx_{st}$.

Theorem 3. *Let $\mathcal{K} = \langle S, AP, L, \longrightarrow\rangle$ be a Kripke structure, $\equiv$ be an equivalence relation on S. If $\equiv$ is an ω-bisimulation, then $\equiv$ is divergence preserving.*

Proof: Suppose $s \equiv t$ and $s \Uparrow_\equiv$. Then there is a divergent run σ from s. Since $\equiv$ is an ω-bisimulation, there exists an infinite run ρ from t such that $(\sigma, \rho) \in \equiv^\sharp$. Then from the condition that σ is a divergent run, it is easy to see that ρ must be a divergent run, thus $t \Uparrow_\equiv$. □

Corollary 1. *$\approx_{st}$ is divergence preserving.*

Proof: Follows immediately from Theorems 2 and 3. □

The following theorem is pretty straight forward.

Theorem 4. *Let $\mathcal{K} = \langle S, AP, L, \longrightarrow\rangle$ be a Kripke structure, R be a binary relation on S. If R is an ω-bisimulation, then R is a stuttering bisimulation.*

Proof: Let $(s,t) \in R$ and $s \longrightarrow s'$, and σ be an infinite run with s, s' as its first two states (since we assume that $\longrightarrow$ is total, such a σ can always be found). Then because R is an ω-bisimulation, there is an infinite run ρ from t such that $(\sigma, \rho) \in R^{\sharp}$. By Definition 2, for ss', which is a finite prefix of σ, there is a finite prefix ρ' of ρ such that $(ss', \rho') \in R^{\natural}$. Thus R is a stuttering bisimulation. □

Corollary 2. *$\approx_{st}$ is a stuttering bisimulation.*

Proof: Follows immediately from the above theorem and Theorem 2. □

From Corollaries 1 and 2, it is easy to see that $\approx_{st}$ is a divergence preserving stuttering bisimulation. In fact this gives an alternative characterization of $\approx_{st}$ which we will prove in the next section: $\approx_{st}$ is the weakest equivalence which is a divergence preserving equivalence and a stuttering bisimulation.

3 Stuttering Bisimulation with Induction

Although the notion of ω-bisimulation makes stuttering equivalence quite straightforward both conceptually and intuitively, it is not very helpful in verification. This is because Definition 5 requires one to examine conditions on infinite runs, of which there are obviously too many to handle in actual verification. What we need is a characterization which can be useful in verification, something like stuttering bisimulation – the conditions to check only concern finite runs of length one. Then the following definition comes into view.

Definition 6 *(*Stuttering bisimulation with induction*). Let $\mathcal{K} = \langle S, AP, L, \longrightarrow \rangle$ be a Kripke structure. For a binary relation $R \subseteq S \times S$, let $\mathcal{B}_I(R)$ be the binary relation inductively defined by the following rule: for $s, t \in S$, if the following hold then $(s,t) \in \mathcal{B}_I(R)$:*

1. *whenever $s \longrightarrow s'$ then either there exists a finite run ρ from t such that $length(\rho) > 0$ and $(ss', \rho) \in R^{\natural}$, or $(s', t) \in R$ and $(s', t) \in \mathcal{B}_I(R)$;*
2. *whenever $t \longrightarrow t'$ then either there exists a finite run σ from s such that $length(\sigma) > 0$ and $(\sigma, tt') \in R^{\natural}$, or $(s, t') \in R$ and $(s, t') \in \mathcal{B}_I(R)$.*

If $R \subseteq \mathcal{B}_I(R)$, then we call R a stuttering bisimulation with induction*. We write $s \approx_{si} t$ if there is a stuttering bisimulation with induction R such that $(s,t) \in R$.*

Comparing the above definition with Definition 3 for stuttering bisimulation, we can find obvious similarities. The rationale behind this definition is as follows. Since $\approx_{st}$ is strictly stronger than $\approx_{db}$, and since those non-divergence preserving pairs in $\approx_{db}$ are extras for $\approx_{st}$, a natural idea to make a stuttering bisimulation like definition for $\approx_{st}$ is to strengthen the conditions of stuttering bisimulation in such a way that those non-divergence preserving pairs are excluded. The definition of stuttering bisimulation with induction did exactly that.

By using the set of ordinals $\mathcal{O}$, the following characterization of $\mathcal{B}_I(R)$ is very helpful in some of the later proofs as well as in understanding the definition of $\mathcal{B}_I(R)$.

Definition 7. *Let $\mathcal{K} = \langle S, AP, L, \longrightarrow\rangle$ be a Kripke structure, R be a binary relation on S. We define $\mathcal{B}_I^\lambda(R)$ for each ordinal $\lambda \in \mathcal{O}$, as follows:*

1. $\mathcal{B}_I^0(R) = \emptyset$.
2. $(s,t) \in \mathcal{B}_I^{\kappa+1}(R)$ *if and only if the following hold:*
   (a) *whenever $s \longrightarrow s'$ then either there exists a finite run ρ from t such that $length(\rho) > 0$ and $(ss', \rho) \in R^\natural$, or $(s', t) \in R$ and $(s', t) \in \mathcal{B}_I^\kappa(R)$;*
   (b) *whenever $t \longrightarrow t'$ then either there exists a finite run σ from s such that $length(\sigma) > 0$ and $(\sigma, tt') \in R^\natural$, or $(s, t') \in R$ and $(s, t') \in \mathcal{B}_I^\kappa(R)$.*
3. *For limit ordinal λ, $(s,t) \in \mathcal{B}_I^\lambda(R)$ if and only if $(s,t) \in \mathcal{B}_I^\kappa(R)$ for some $\kappa < \lambda$.*

Theorem 5. *Let $\mathcal{K} = \langle S, AP, L, \longrightarrow\rangle$ be a Kripke structure, R be a binary relation on S. Then for $s, t \in S$ the following hold:*

1. $\mathcal{B}_I(R) = \bigcup_{\lambda\in\mathcal{O}} \mathcal{B}_I^\lambda(R)$;
2. *if λ is the least ordinal with $(s,t) \in \mathcal{B}_I^\lambda(R)$, then $\lambda = \kappa + 1$ for some $\kappa \in \mathcal{O}$.*

Proof. 1 can be proved by standard fixed-point arguments. To see 2, just note that λ cannot be 0 since $\mathcal{B}_I^0(R)$ is empty, neither can it be a limit since otherwise there would exist a smaller ordinal κ with $(s,t) \in \mathcal{B}_I^\kappa(R)$. □

In the rest of this section, our major task is to prove that the resulting relation $\approx_{si}$ is indeed the same as $\approx_{st}$.

Lemma 2. *If $R_1 \subseteq R_2$, then $\mathcal{B}_I(R_1) \subseteq \mathcal{B}_I(R_2)$.*

Proof. Easy to prove by induction on the definition of $\mathcal{B}_I(R_1)$, or to use the ordinal characterization and prove by induction on all $\lambda \in \mathcal{O}$. □

This lemma shows a very nice property of the definition. It essentially says that if we consider $\mathcal{B}_I$ as a function on binary relations then it is monotonic. Then Knaster-Tarski fixed-point theorem can be applied to the complete lattice $(2^{S\times S}, \subseteq)$ to obtain $\approx_{si}$ as the maximum fixed-point of $\mathcal{B}_I$.

Theorem 6. *$\approx_{si}$ is a stuttering bisimulation with induction, and it is the largest stuttering bisimulation with induction, and moreover $\approx_{si} = \mathcal{B}_I(\approx_{si})$.*

Proof. The theorem is in fact an instance of Knaster-Tarski fixed-point theorem. To show that $\approx_{si}$ is a stuttering bisimulation with induction, we have to establish $\approx_{si} \subseteq \mathcal{B}_I(\approx_{si})$. Suppose that R is a stuttering bisimulation with induction, then obviously $R \subseteq \mathcal{B}_I(R)$ and $R \subseteq \approx_{si}$. According to Lemma 2, $\mathcal{B}_I$ is monotonic, thus $\mathcal{B}_I(R) \subseteq \mathcal{B}_I(\approx_{si})$, so we showed that for any stuttering bisimulation with induction R it holds that $R \subseteq \mathcal{B}_I(\approx_{si})$. Now to see $\approx_{si} \subseteq \mathcal{B}_I(\approx_{si})$, just note that $\approx_{si} = \bigcup\{R \mid R \text{ is a stuttering bisimulation with induction}\}$.

If R is a stuttering bisimulation with induction, then by the definition obviously $R \subseteq \approx_{si}$. Thus $\approx_{si}$ is the largest stuttering bisimulation with induction.

We have just shown above that $\approx_{si} \subseteq \mathcal{B}_I(\approx_{si})$, then since $\mathcal{B}_I$ is monotonic, $\mathcal{B}_I(\approx_{si}) \subseteq \mathcal{B}_I(\mathcal{B}_I(\approx_{si}))$. So $\mathcal{B}_I(\approx_{si})$ is a stuttering bisimulation with induction, thus $\mathcal{B}_I(\approx_{si}) \subseteq \approx_{si}$ and $\mathcal{B}_I(\approx_{si}) = \approx_{si}$. □

Remark 1. It is clear from this theorem that $\approx_{si}$ is the greatest fixed-point of $\mathcal{B}_I$, i.e. $\approx_{si} = \nu R(\mathcal{B}_I(R))$ in μ-calculus notation. In fact, from Definition 6, it is also clear that $\mathcal{B}_I(R)$ itself is the least fixed-point of $\mathcal{F}(R)$, where for a given $R^* \subseteq S \times S$, $(s,t) \in \mathcal{F}(R)(R^*)$ if and only if the following hold:

1. whenever $s \longrightarrow s'$ then either there exists a finite run ρ from t such that $length(\rho) > 0$ and $(ss', \rho) \in R^\natural$, or $(s',t) \in R$ and $(s',t) \in R^*$;
2. whenever $t \longrightarrow t'$ then either there exists a finite run σ from s such that $length(\sigma) > 0$ and $(\sigma, tt') \in R^\natural$, or $(s,t') \in R$ and $(s,t') \in R^*$.

Then $\approx_{si} = \nu R(\mu R^*(\mathcal{F}(R)(R^*)))$, that is, $\approx_{si}$ is expressed as an alternating fixed-point (it is easy to see that $\mathcal{F}(R)$ is monotonic, thus the least fixed-point is well defined). For relations defined as an alternating fixed-point, there are efficient local algorithms to decide whether $(s,t) \in \nu R(\mu R^*(\mathcal{F}(R)(R^*)))$ where s,t are states of a Kripke structure with finite states set, see e.g. [10]. Here the localness means that the algorithm does not compute the whole of $\nu R(\mu R^*(\mathcal{F}(R)(R^*)))$ in order to decide whether $(s,t) \in \nu R(\mu R^*(\mathcal{F}(R)(R^*)))$ holds, it only computes a part $P \subseteq \nu R(\mu R^*(\mathcal{F}(R)(R^*)))$ which is big enough to decide whether $(s,t) \in \nu R(\mu R^*(\mathcal{F}(R)(R^*)))$ holds. In fact such P is just stuttering bisimulation with induction. Thus the characterization of stuttering equivalence in stuttering bisimulation with induction facilitates local decision strategy, which would give stuttering bisimulation with induction a clear advantage in verification practice.

Another important property to establish about $\approx_{si}$ is that it is an equivalence relation. Unfortunately, it is not an easy task to directly prove that $\approx_{si}$ is transitive. Here we will take an indirect approach, since anyhow we are going to establish that $\approx_{si} = \approx_{st}$ (Theorem 9). Then from the fact that $\approx_{st}$ is an equivalence relation, we immediately know that so is $\approx_{si}$.

Theorem 7. *Let $\equiv$ be an equivalence. If $\equiv$ is divergence preserving, and is a stuttering bisimulation, then $\equiv \subseteq \mathcal{B}_I(\equiv)$.*

Proof. First define a binary relation $\succ \subseteq S \times S$ such that $s \succ s'$ if and only if s is not divergent and $s \equiv s'$ and $s \longrightarrow s'$. Then it is clear that if $\equiv$ is divergence preserving, then $\succ$ is well founded, i.e. there is no infinite descending chain $s \succ s_1 \ldots \succ s_i \ldots$. Otherwise $\sigma = ss_1 \ldots s_i \ldots$ would be a divergent run from s.

Now suppose that $\equiv$ is a stuttering bisimulation, and $s \equiv t$, we will show $(s,t) \in \mathcal{B}_I(\equiv)$ by well-founded induction on $\succ$. Let $s \longrightarrow s'$ be any transition from s, we have to find a match for it that meets the requirements in Definition 6. Since $\equiv$ is a stuttering bisimulation, $s \equiv t$, then there must exist a finite run ρ from t such that $(ss', \rho) \in R^\natural$. Now we can discuss in two cases. The first case is that we can find such a ρ with $length(\rho) > 0$, then a required match for $s \longrightarrow s'$ is found. The second case is that, the only such ρ has length 0, and in this case $s' \equiv t$. Obviously t must not be a divergent state (otherwise there is a divergent run η from t, and any finite prefix ρ of η satisfies $(ss', \rho) \in R^\natural$), and since $\equiv$ is divergence preserving, then s is not divergent. Now $s \succ s'$, $s' \equiv t$, and by the induction hypothesis $(s',t) \in \mathcal{B}_I(\equiv)$, and a required match for $s \longrightarrow s'$ is also found. Thus we proved $(s,t) \in \mathcal{B}_I(\equiv)$. □

Corollary 3. *$\approx_{st}$ is a stuttering bisimulation with induction, and $\approx_{st} \subseteq \approx_{si}$.*

Proof. From Corollaries 1, and 2, $\approx_{st}$ is a divergence preserving equivalence and it is a stuttering bisimulation. Then by Theorem 7 $\approx_{st} \subseteq \mathcal{B}_I(\approx_{st})$, thus $\approx_{st}$ is a stuttering bisimulation with induction, and $\approx_{st} \subseteq \approx_{si}$. □

To establish the other direction, we need to show that $\approx_{si}$ is an ω-bisimulation.

Lemma 3. *Let $\mathcal{K} = \langle S, AP, L, \longrightarrow \rangle$ be a Kripke structure, R be a binary relation on S. For all $\lambda \in \mathcal{O}$, if $(s,t) \in \mathcal{B}_I^{\lambda}(R)$ then the following hold*

1. *if σ is an infinite run from s, then there is a finite run ρ from t with $length(\rho) > 0$, and a finite prefix σ^* of σ such that $(\sigma^*, \rho) \in R^{\natural}$;*
2. *if ρ is an infinite run from t, then there is a finite run σ from s with $length(\sigma) > 0$, and a finite prefix ρ^* of ρ such that $(\sigma, \rho^*) \in R^{\natural}$.*

Proof. Here we only show 1, because 2 can be proved in the same way. We prove by induction on $\lambda \in \mathcal{O}$. If $\lambda = 0$ there is nothing to be proved. If $\lambda = \kappa + 1$, let $(s,t) \in \mathcal{B}_I^{\kappa+1}(R)$, $\sigma = s_1 s_2 s_3 \ldots$. Then $s_1 \longrightarrow s_2$, according to the definition of $\mathcal{B}_I^{\kappa+1}(R)$, there are the following two cases. The first case is that there exists a finite run ρ from t such that $length(\rho) > 0, (s_1 s_2, \rho) \in R^{\natural}$, and in this case take $s_1 s_2$ as σ^*, then ρ is the required run from t. The second case is that $(s_2, t) \in R$ and $(s_2, t) \in \mathcal{B}_I^{\kappa}(R)$, and in this case by the induction hypothesis, for the infinite run $\sigma' = s_2 s_3 \ldots$, there is a finite run ρ from t with $length(\rho) > 0$, and there is a finite prefix $\sigma^{\dagger}$ of σ' such that $(\sigma^{\dagger}, \rho) \in R^{\natural}$, and in this case we take $\sigma^* = s_1 \sigma^{\dagger}$, then $(\sigma^*, \rho) \in R^{\natural}$ and ρ is the required run. If λ is a limit ordinal, then there is $\kappa \in \mathcal{O}$ such that $\kappa < \lambda$ and $(s,t) \in \mathcal{B}_I^{\kappa}(R)$, then the induction hypothesis immediately gives a finite run ρ from t with $length(\rho) > 0$, and there is a finite prefix σ^* of σ such that $(\sigma^*, \rho) \in R^{\natural}$. □

Theorem 8. *Let $\mathcal{K} = \langle S, AP, L, \longrightarrow \rangle$ be a Kripke structure, R be a binary relation on S. If R is a stuttering bisimulation with induction, then R is an ω-bisimulation.*

Proof. Suppose $R \subseteq \mathcal{B}_I(R)$, and $(s,t) \in R$ we need to prove the following:

1. if σ is an infinite run from s, then there is an infinite run ρ from t such that $(\sigma, \rho) \in R^{\sharp}$;
2. if ρ is an infinite run from t, then there is an infinite run σ from s such that $(\sigma, \rho) \in R^{\sharp}$.

Here we only prove 1, because 2 can be proved in the same way. So suppose σ is an infinite run from s. Since in this case $(s,t) \in \mathcal{B}_I(R)$, then $(s,t) \in \mathcal{B}_I^{\lambda}(R)$ for some $\lambda \in \mathcal{O}$, by Lemma 3 we can obtain a finite prefix σ_1 of σ and a finite run ρ_1 from t with $length(\rho_1) > 0$, such that $(\sigma_1, \rho_1) \in R^{\natural}$. Now we can do the same thing for $(last(\sigma_1), last(\rho_1)) \in R$ with the infinite run which is the remaining part of σ after σ_1. Repeating the process to infinity we obtain a run ρ by concatenating $\rho_1, \rho_2, \ldots$ in the obvious way. Since each ρ_i has a positive length, clearly ρ is an infinite run and it is not difficult to see that $(\sigma, \rho) \in R^{\sharp}$. □

Finally, we are ready to prove:

Theorem 9. $\approx_{si}=\approx_{st}$.

Proof. According to Theorem 6, $\approx_{si}$ is a stuttering bisimulation with induction, then, by Theorem 8, $\approx_{si}$ is an ω-bisimulation, thus $\approx_{si}\subseteq\approx_{st}$. Then combine this with Corollary 3 we obtain $\approx_{si}=\approx_{st}$. □

Thus $\approx_{si}$ is an equivalence relation. As we promised in the end of the last section, we have to prove the following important characterization of $\approx_{si}$ and $\approx_{st}$.

Theorem 10. *$\approx_{st}$ (as well as $\approx_{si}$) is the weakest equivalence which is a stuttering bisimulation and at the same time is a divergence preserving equivalence.*

Proof. Now it is clear that $\approx_{st}$ is divergence preserving and is a stuttering bisimulation. To show that it is the weakest such, let $\equiv$ be a stuttering bisimulation and at the same time it is a divergence preserving equivalence. Then by Theorem 7 $\equiv\subseteq \mathcal{B}_I(\equiv)$, thus $\equiv\subseteq\approx_{si}=\approx_{st}$. □

Remark 2. In principle, one can "define" an equivalence relation $\simeq$ by requiring that $s \simeq t$ if and only if there exists a divergence preserving equivalence relation $\equiv$ such that $\equiv$ is a stuttering bisimulation and $s \equiv t$. However, such kind of "definition" needs to be justified in order to be meaningful. In particular, one needs to prove that the defined relation $\simeq$ is indeed an equivalence relation, and is divergence preserving, and is a stuttering bisimulation. Here Theorem 10 provides the justification for $\simeq$. In some cases this kind of justification is routine. Such examples include strong and weak bisimulation equivalences, branching bisimulation equivalence, etc. In these examples, due to the existence of obvious monotonic functions, application of Knaster-Tarski fixed-point theorem turned the justification into a routine task. In other cases it cannot be considered routine, where justification is difficult by the definition itself, and one needs to find other ways to get around. This is the case here, since from the definition itself it is not obvious how to prove that $\simeq$ is an equivalence relation, one has to construct an equivalence by other means (like $\approx_{st}$ or $\approx_{si}$), and then to use that to prove that $\simeq$ is an equivalence. It is for this reason that we do not consider the way of introducing $\simeq$ as desirable. It easily causes confusion while does not save any amount of work.

4 Well-Founded Bisimulation

In [11], the notion of well-founded bisimulation was proposed to capture stuttering equivalence. In this section we study its relationship to stuttering bisimulation with induction.

Definition 8 *(*Well-founded bisimulation*). Let $\mathcal{K} = \langle S, AP, L, \longrightarrow\rangle$ be a Kripke structure. Let $rank : S \times S \times S \to W$ be a total function, where $(W, \prec)$ is well-founded. A binary relation $R \subseteq S \times S$ is a well-founded bisimulation w.r.t. rank iff R is symmetric and for every $(s, t) \in R$ the following hold:*

1. $L(s) = L(t)$;
2. *whenever* $s \longrightarrow u$ *then one of the following must hold:*
   (a) $t \longrightarrow v$ *for some* $v \in S$ *with* $(u, v) \in R$;
   (b) $(u, t) \in R$ *and* $rank(u, u, t) \prec rank(s, s, t)$;
   (c) $(u, t) \notin R$ *and* $t \longrightarrow v$ *for some* $v \in S$ *with* $(s, v) \in R$ *and* $rank(u, s, v) \prec rank(u, s, t)$.

The purpose of the ternary rank function $rank(u, s, t)$, when used in case (c) in the above definition, is to enforce an order in defining the condition that the transition $s \longrightarrow u$ can be matched by a transition from t. In case (b), the rank function is used to enforce an order in defining the condition that the transition $s \longrightarrow u$ can be matched by default. In principle the two well founded orders in case (b) and case (c) of the definition are different: the former being an order between pairs of states and latter an order between triples of states. It is just a coincidence that the function *rank* can serve both purpose.

First, we show that every well-founded bisimulation is a stuttering bisimulation with induction.

Theorem 11. *Let* $\mathcal{K} = \langle S, AP, L, \longrightarrow \rangle$ *be a Kripke structure. If* R *is a well-founded bisimulation on* $\mathcal{K}$ *with some well-founded set* $(W, \prec)$ *and total function rank, then* R *is a stuttering bisimulation with induction.*

Proof. We first establish the following fact by well-founded induction on $\prec$:

> If $(s, t) \in R$ and $s \longrightarrow u$ and $(u, t) \notin R$, then there exists a finite run ρ such that $length(\rho) > 0$ and $(su, \rho) \in R^\natural$.

To show that, suppose $(s, t) \in R$ and $s \longrightarrow u$ and $(u, t) \notin R$. Since R is a well-founded bisimulation, one of the conditions in $(a), (b), (c)$ of Definition 8 must hold. However because $(u, t) \notin R$, condition (b) is excluded, thus either (a) or (c) must hold. If (a) holds, then $t \longrightarrow v$ with $(u, v) \in R$ for some $v \in S$, clearly tv is the ρ we are looking for. If (c) holds, then $t \longrightarrow v$ for some $v \in S$ with $(s, v) \in R$ and $rank(u, s, v) \prec rank(u, s, t)$. Now we have two subcases to discuss: $(u, v) \in R$ and $(u, v) \notin R$. In the first subcase, again tv is the ρ we are looking for. In the second subcase, because $(s, v) \in R$, $s \longrightarrow u$, $(u, v) \notin R$, and $rank(u, s, v) \prec rank(u, s, t)$, by the induction hypothesis there is a finite run ρ' from v such that $length(\rho') > 0$ and $(su, \rho') \in R^\natural$. Let $\rho = t\rho'$, clearly $(su, \rho) \in R^\natural$.

Now suppose $(s, t) \in R$, we show $(s, t) \in \mathcal{B}_I(R)$ by well-founded induction as follows. Let $s \longrightarrow u$, then we have two cases to discuss: $(u, t) \notin R$ and $(u, t) \in R$. In the first case, by the fact we proved above there is a finite run ρ from t with $length(\rho) > 0$ and $(su, \rho) \in R^\natural$. In the second case, by the condition that R is a well-founded bisimulation, one of the conditions in $(a), (b), (c)$ of Definition 8 must hold. However (c) is clearly excluded because in this case $(u, t) \in R$. So we have two subcases to discuss. If (a) holds, then $t \longrightarrow v$ with $(u, v) \in R$ for some $v \in S$, clearly $(su, tv) \in R^\natural$. If (b) holds, then $(u, t) \in R$ and $rank(u, u, t) \prec rank(s, s, t)$, by the induction hypothesis $(u, t) \in \mathcal{B}_I(R)$. Thus,

for the given (s,t), we showed that whenever $s \longrightarrow u$ then either there is a finite run ρ from t such that $length(\rho) > 0$ and $(su,\rho) \in R^\natural$, or $(u,t) \in R$ and $(u,t) \in \mathcal{B}_I(R)$, hence $(s,t) \in \mathcal{B}_I(R)$. □

Next, we show that a symmetric stuttering bisimulation with induction is a well-founded bisimulation.

Theorem 12. *Let $\mathcal{K} = \langle S, AP, L, \longrightarrow\rangle$ be a Kripke structure, R be a symmetric binary relation on S. If R is a stuttering bisimulation with induction, then R is a well-founded bisimulation with some well-founded set $(W, \prec)$ and total function rank.*

Proof. Define $rank : S \times S \times S \to \mathcal{O}$ as follows:

1. For $s,t \in S$, if $(s,t) \in \mathcal{B}_I(R)$ then $rank(s,s,t) = \lambda$ where λ is the least ordinal such that $(s,t) \in \mathcal{B}_I^\lambda(R)$ (by Theorem 5 $\lambda = \kappa + 1$ for some $\kappa \in \mathcal{O}$), if $(s,t) \notin \mathcal{B}_I(R)$ then $rank(s,s,t) = 0$;
2. For $u,s,t \in S$ with u,s being two different states, then $rank(u,s,t) = l$, where l is the length of the shortest finite run ρ from t such that $length(\rho) > 0$ and $(su,\rho) \in R^\natural$ if such a ρ exists, otherwise let $l = 0$ (in fact just let l be any value will do in this case).

Suppose R is a stuttering bisimulation with induction and R is symmetric, we will show that with $(\mathcal{O}, <)$ and $rank$ defined above, R is a well-founded bisimulation. Let $(s,t) \in R$. Since R is a stuttering bisimulation with induction, thus $(s,t) \in \mathcal{B}_I(R)$, and let $rank(s,s,t) = \lambda = \kappa + 1$, so $(s,t) \in \mathcal{B}_I^{\kappa+1}(R)$. First note that in this case $L(s) = L(t)$. Suppose $s \longrightarrow u$, since $(s,t) \in \mathcal{B}_I^{\kappa+1}(R)$, two of the following will happen. Either there exists a finite run ρ from t such that $length(\rho) > 0$ and $(su,\rho) \in R^\natural$, or $(u,t) \in R$ and $(u,t) \in \mathcal{B}_I^\kappa(R)$. In the latter case, obviously $rank(u,u,t) < \lambda$, so condition (b) of Definition 8 is satisfied. In the former case, according to the definition of $rank(u,s,t)$, there is $l > 0$ and $rank(u,s,t) = l$. We distinguish two subcases. The first subcase is that $l = 1$, then let $\rho = tv$, and clearly $t \longrightarrow v, (u,v) \in R$, condition (a) of Definition 8 is satisfied. The second subcase is that $l > 1$, then according to the definition of $rank(u,s,t)$ there is $\rho = tvv_1 \ldots$ which is the shortest finite run from t such that $length(\rho) = l$ and $(su,\rho) \in R^\natural$. With a detailed case analysis it is not difficult to see that $(s,v) \in R$, and $rank(u,s,v) \leq l - 1 < rank(u,s,t)$, and condition (c) of Definition 8 is satisfied. □

Theorems 11 and 12 not only imply that well-founded bisimulation and (symmetric) stuttering bisimulation with induction both characterize the same relation, i.e. stuttering equivalence, but also claim that the two notions are essentially the same thing. From the point of view of verification practice, each of the two notions has its own advantages. As explained in *Remark* 1, stuttering bisimulation with induction is presented as an alternating fixed-point of some monotonic function on the complete lattice of binary relations, thus existing efficient local correctness checking strategies can be applied to decide whether some given pairs of states are stuttering equivalent. With given well-founded set and function $rank$, the conditions of well-founded bisimulation is easy to verify.

So a well-founded bisimulation R can be used as a proof that the pairs in the relation are stuttering equivalent. In other words, stuttering bisimulation with induction is more useful for equality checking, while well-founded bisimulation is better suited for equality proving. In fact the two can be combined in such a way that first using a fast local algorithm to obtain a relation R which is a stuttering bisimulation with induction, and then using the construction in the proof of Theorem 12 on the symmetric stuttering bisimulation $R \cup R^{-1}$ to obtain a well-founded bisimulation as a proof for the equality of all pairs in R. In fact the construction can be turned into an algorithm which, for a given symmetric stuttering bisimulation with induction R on a Kripke structure with finite number of states, computes the function *rank* for the well-founded bisimulation. We describe such an algorithm in the rest of this section.

To describe the algorithm, we need a theorem which says that often it is sufficient to stay out of limit ordinals. The following lemma is needed for proving the theorem.

Lemma 4. *If $\kappa < \lambda$ then $\mathcal{B}_I^\kappa(R) \subseteq \mathcal{B}_I^\lambda(R)$.*

The proof of the lemma is standard, and we omit it here.

Theorem 13. *Let $\mathcal{K} = \langle S, AP, L, \longrightarrow \rangle$ be a Kripke structure, R be a binary relation on S.*

1. *If $\longrightarrow$ is finite branching, i.e. $\{s \in S \mid s_0 \longrightarrow s\}$ is a finite set for all $s_0 \in S$, then whenever $(s,t) \in \mathcal{B}_I(R)$ there is a natural number n such that $(s,t) \in \mathcal{B}_I^n(R)$.*
2. *If S is finite with m states, then there exists n with $0 < n \leq m^2$ such that $\mathcal{B}_I^0(R) \subseteq \mathcal{B}_I^1(R) \ldots \subseteq \mathcal{B}_I^n(R)$ is an increasing chain and $\mathcal{B}_I(R) = \mathcal{B}_I^n(R)$.*

Proof. To prove 1, it is sufficient to prove by induction that in this case for all $\lambda \in \mathcal{O}$ if $(s,t) \in \mathcal{B}_I^\lambda(R)$ then there is a natural number n such that $(s,t) \in \mathcal{B}_I^n(R)$. If λ is a natural number, then the claim trivially holds. If λ is a limit ordinal, by Definition 7 there is $\kappa < \lambda$ such that $(s,t) \in \mathcal{B}_I^\kappa(R)$, then by the induction hypothesis there is a natural number n such that $(s,t) \in \mathcal{B}_I^n(R)$. If $\lambda = \kappa + 1$, by Definition 7 and the induction hypothesis the following hold:

1. whenever $s \longrightarrow s'$ then either there exists a finite run ρ from t such that $length(\rho) > 0$ and $(ss', \rho) \in R^\natural$, or $(s', t) \in R$ and $(s', t) \in \mathcal{B}_I^n(R)$ for some natural number n;
2. whenever $t \longrightarrow t'$ then either there exists a finite run σ from s such that $length(\sigma) > 0$ and $(\sigma, tt') \in R^\natural$, or $(s, t') \in R$ and $(s, t') \in \mathcal{B}_I^n(R)$ for some natural number n.

Now since $\longrightarrow$ is finite branching, $\{s' \mid s \longrightarrow s'\} \cup \{t' \mid t \longrightarrow t'\}$ is a finite set, we can choose the maximum among the finitely many n's, and let it be m, then by Lemma 4 the following hold

1. whenever $s \longrightarrow s'$ then either there exists a finite run ρ from t such that $length(\rho) > 0$ and $(ss', \rho) \in R^\natural$, or $(s', t) \in R$ and $(s', t) \in \mathcal{B}_I^m(R)$;

2. whenever $t \longrightarrow t'$ then either there exists a finite run σ from s such that $length(\sigma) > 0$ and $(\sigma, tt') \in R^{\natural}$, or $(s, t') \in R$ and $(s, t') \in \mathcal{B}_I^m(R)$.

So in this case $(s, t) \in \mathcal{B}^{m+1}(R)$.

In order to prove 2, note that when S has m elements, the size of the relations in the increasing chain $\mathcal{B}_I^0(R) \subseteq \mathcal{B}_I^1(R) \ldots \subseteq \mathcal{B}_I^n(R) \ldots$ is bounded by m^2. So there exists n with $0 < n \leq m^2$ such that $\mathcal{B}_I^n(R) = \mathcal{B}_I^{n+1}(R)$. Then it is easy to prove by induction that for all $\lambda \in \mathcal{O}$, it holds that $\mathcal{B}_I^{\lambda}(R) \subseteq \mathcal{B}_I^n(R)$. Then $\mathcal{B}_I(R) = \mathcal{B}_I^n(R)$ follows easily. □

By Theorem 13, we know that when the Kripke structure is finite branching, in constructing *rank* for the well-founded bisimulation in Theorem 12, we can always use natural numbers as the well-founded set for the well-founded bisimulation. And when the Kripke structure has only finite number of states, we can always use a finite subset of natural numbers as the well-founded set for the well-founded bisimulation, and moreover in this case there is n such that $\mathcal{B}_I(R) = \mathcal{B}_I^n(R)$.

We assume a basic procedure FindRun which takes (u, s, t) as input where $(s, t) \in R$ and $s \longrightarrow u$, and find the shortest run ρ from t such that $length(\rho) > 0$ and $(su, \rho) \in R^{\natural}$. It outputs the length of such a run if there exists one, or it outputs 0. It is not difficult to see that this is similar to looking for the shortest path in a graph, which can be implemented with time complexity polynomial to the size of the state set.

Now, for a given symmetric stuttering bisimulation R, the algorithm constructs a well-founded bisimulation as follows.

First, according to Definition 7, we use FindRun to compute $\mathcal{B}_I^k(R)$ from $k = 0$ until $k = n$ where $\mathcal{B}_I^n(R) = \mathcal{B}_I^{n+1}(R)$. It is not difficult to see that each $\mathcal{B}_I^k(R)$ can be computed with time polynomial to the size of the state set. Thus the overall time complexity for computing $\mathcal{B}_I^k(R)$ from $k = 0$ to $k = n$ is also polynomial to the size of the state set.

As the last step, we construct *rank* as follows:

1. for (u, s, t) with u, s being different states, $(s, t) \in R$ and $s \longrightarrow u$, let $rank(u, s, t) = l$ where l is the output of FindRun(u, s, t);
2. for (s, s, t) with $(s, t) \in R$, let $rank(s, s, t) = l$ where $(s, t) \in \mathcal{B}_I^l(R)$ and $(s, t) \notin \mathcal{B}_I^{l-1}(R)$;
3. for the rest of (u, s, t), let $rank(u, s, t) = 0$.

It is not difficult to see that the total time complexity of the algorithm is polynomial to the size of the state set. According to the proof of Theorem 12, R with *rank* is a well-founded bisimulation.

5 Related Works

The notion of stuttering bisimulation with induction is an adaptation of the notion of inductive branching bisimulation introduced in [16], which is the study of the labeled transition system version of divergence preserving stuttering equivalence. The presentation of the theory part in Sects. 3 and 4 is slightly different

from the presentation in [16]. In particular, the notion of ω-bisimulation is introduced in place of the complete branching bisimulation. The new presentation is simpler and more concise for stuttering equivalence due to the complete nature of the transition relation in Kripke structures, i.e. for any state there always exists some out-going transition. As an equivalence which has the exact distinguishing strength as the set of CTL*-X properties, stuttering equivalence is certainly very important. However, it seems that it is still in need for a general rigorous formulation. It is for this purpose that we propose ω-bisimulation as a candidate for this role. From the theoretical development in this paper, it looks fit for this role. The formulation in the original paper [3] is in the form of the limit of a convergence sequence of equivalence relations, which is not easy to use in proving theorems about it. Also it is assumed for finite state systems, which makes it not general enough. In [11] the formulation of stuttering equivalence relies on a non-trivial definition of a matching relation which only appears in the appendix of the paper. The matching relation makes the formulation not easy to handle, in particular it seems not easy to establish that the final relation is indeed an equivalence (no proof has been provided in the paper).

The notion of well-founded bisimulation was introduced in [11] and studied in [9] to characterize stuttering equivalence. Due to the lack of a clear theoretical foundation for stuttering equivalence, the characterization proofs in the mentioned works left something to be desired. In [9], stuttering equivalence is characterized by the so-called divergence sensitive stutter bisimulation, which is based on the notion of $\mathcal{R}$-divergence. This characterization is similar to the kind of definition mentioned in *Remark* 2. Although the characterization proof of well-founded bisimulation was provided in [9] in terms of divergence sensitive stutter bisimulation, such characterization of stuttering equivalence itself needs further justification. Theorems 11 and 12 and their proofs in this paper provide a sound theoretical foundation for well-founded bisimulation. Another disadvantage of divergence sensitive stutter bisimulation is, since it relies on equivalence relations that satisfy certain conditions on infinite paths, it would need much effort when it is used directly as a method for proving the equivalence of states.

In [13] branching bisimulation with explicit divergence is studied in detail, which is the labeled transition system version of stuttering equivalence. Branching bisimulation with explicit divergence is defined similar to the kind of definition mentioned in *Remark* 2. In [13] the authors took serious efforts to complete the needed justification for the definition. However the proofs there were quite complicated due to the lack of good co-inductive properties of the definition.

In [15], a partition based efficient algorithm for divergence blind stuttering equivalence was presented. Also a transformation between finite Kripke structures was given in [15] such that two states are divergence blind stuttering equivalent in the transformed Kripke structure if and only if the corresponding states in the original Kripke structure are stuttering equivalent. This implies that the problem of checking stuttering equivalence can also be solved by their algorithm. Compared to the local algorithm approach mentioned in *Remark* 1, the partition based algorithm has better worst case time complexity due to the exploitation of

good properties of equivalence relation. However partition algorithms are inherently global, which makes them unable to exploit early termination chances.

6 Conclusion

In this paper, we propose the notion of stuttering bisimulation with induction to characterize stuttering equivalence. It is argued that, due to its fixed-point style definition, stuttering bisimulation with induction is a good characterization for stuttering equivalence in that there are efficient local algorithms for the equality checking problem. We also use stuttering bisimulation with induction to analyze the notion of well-founded bisimulation. It is shown that stuttering bisimulation with induction and well-founded bisimulation are essentially the same thing, and as a byproduct a method for constructing the ranking function for well-founded bisimulation from a given stuttering bisimulation with induction is presented. Also a notion of ω-bisimulation is introduced to characterize stuttering equivalence, which leads to smooth development of the theory.

As we pointed out in Sect. 4, stuttering bisimulation with induction and well-founded induction are good for different things, the former is good for equality checking and latter for equality proving. In this respect an interesting future work is to combine them in a tool where a local equality decision procedure produces a stuttering bisimulation with induction, which is then used to construct the corresponding well-founded bisimulation. And the resulting well-founded bisimulation can act as a proof of equality for the elements in the stuttering bisimulation with induction.

References

1. Hennessy, M.C.B., Plotkin, G.D.: A term model for CCS. In: Dembiński, P. (ed.) MFCS 1980. LNCS, vol. 88, pp. 261–274. Springer, Heidelberg (1980). https://doi.org/10.1007/BFb0022510
2. Walker, D.J.: Bisimulation and divergence. Inf. Comput. **85**, 212–241 (1990)
3. Browne, M.C., Clarke, E.M., Grümberg, O.: Characterizing finite Kripke structures in propositional temporal logic. Theor. Comput. Sci. **59**, 115–131 (1988)
4. Browne, M.C., Clarke, E.M., Grumberg, O.: Reasoning about networks with many identical finite state processes. Inf. Comput. **81**(1), 13–31 (1989)
5. Glabbeek, R.J.: The linear time — branching time spectrum II. In: Best, E. (ed.) CONCUR 1993. LNCS, vol. 715, pp. 66–81. Springer, Heidelberg (1993). https://doi.org/10.1007/3-540-57208-2_6
6. de Nicola, R., Vaandrager, F.: Three logics for branching bisimulation. J. ACM **42**(2), 458–487 (1995)
7. Park, D.: Concurrency and automata on infinite sequences. In: Deussen, P. (ed.) GI-TCS 1981. LNCS, vol. 104, pp. 167–183. Springer, Heidelberg (1981). https://doi.org/10.1007/BFb0017309
8. Milner, R.: Communication and Concurrency. Prentice-Hall, New York (1989)
9. Baier, C., Katoen, J.-P.: Principles of Model Checking. The MIT Press, Cambridge (2008)

10. Vergauwen, B., Lewi, J.: Efficient local correctness checking for single and alternating boolean equation systems. In: Abiteboul, S., Shamir, E. (eds.) ICALP 1994. LNCS, vol. 820, pp. 304–315. Springer, Heidelberg (1994). https://doi.org/10.1007/3-540-58201-0_77
11. Namjoshi, K.S.: A simple characterization of stuttering bisimulation. In: Ramesh, S., Sivakumar, G. (eds.) FSTTCS 1997. LNCS, vol. 1346, pp. 284–296. Springer, Heidelberg (1997). https://doi.org/10.1007/BFb0058037
12. van Glabbeek, R.J., Weijland, P.: Branching time and abstraction in bisimulation semantics. J. ACM **43**(3), 555–600 (1996)
13. van Glabbeek, R.J., Luttik, B., Trcka, N.: Branching bisimilarity with explicit divergence. Fundam. Inform. **93**(4), 371–392 (2009)
14. Allen Emerson, E., Halpern, J.Y.: "Sometimes" and "Not Never" revisited: on branching versus linear time temporal logic. J. ACM **33**(1), 151–178 (1986)
15. Groote, J.F., Vaandrager, F.: An efficient algorithm for branching bisimulation and stuttering equivalence. In: Paterson, M.S. (ed.) ICALP 1990. LNCS, vol. 443, pp. 626–638. Springer, Heidelberg (1990). https://doi.org/10.1007/BFb0032063
16. Liu, X., Yu, T., Zhang, W.: Analyzing divergence in bisimulation semantics. In: Proceedings of 44th ACM SIGPLAN Symposium on Principles of Programming Languages (POPL2017), Paris (2017)