STUTTERING CONGRUENCE FOR χ

Bas Luttik^{1,2} and Nikola Trčka¹

¹ Department of Mathematics and Computer Science, Eindhoven University of Technology, P.O. Box 513, NL-5600 MB Eindhoven, The Netherlands

² CWI, P.O. Box 94079, NL-1090 GB Amsterdam, The Netherlands

Abstract. The language χ is a modeling and simulation language which is currently mainly used to analyse and optimize the performance of industrial systems. To be able to also verify functional properties of a system using a χ model, part of the language has been given a formal semantics. Rather than implementing a new model checker for χ , the philosophy is to provide automatic translations from χ into the specification languages of existing state-of-the-art model checkers such as, e.g., SPIN and UPPAAL.

In this paper, we propose for χ a notion of stuttering congruence, which is an adaptation of the notion of stuttering equivalence. We prove that our notion preserves the validity of CTL^*_χ formulas, that it preserves deadlock, and that it is indeed a congruence with respect to the constructs of χ . We also indicate how our notion is to be used to establish confidence in the correctness of a translation from χ into PROMELA.

1 Introduction

The language χ [15] is a modeling language developed for detecting design flaws and for optimizing performance of industrial systems (machines, manufacturing lines, warehouses, factories, etc.) by simulation. Quite a few case studies have shown the usefulness of χ in an industrial context [11, 6, 9, 17]; simulation turns out to be a powerful technique for doing performance analysis such as approximating throughput and cycle time. However, for the verification of functional properties such as, e.g., deadlock freedom, simulation is less suitable. To be able to also do verification with χ , either verification tools have to be developed especially for χ , or existing state-of-the-art verification tools and techniques have to be made available for use with χ . Currently, the latter approach is pursued. The idea is to extend χ with facilities for doing formal verification by establishing a connection with other state-of-the-art verification tools and techniques on the level of the specification language. That is, formal verification of a χ model is done by first translating it into the input language of some model checker and then performing the actual verification. The suitability of this approach was shown in [2, 3], where a χ model of a turntable machine was translated to PROMELA [12], μCRL [1] and UPPAAL timed automata [13], and then verified in SPIN, CADP [8] and UPPAAL, respectively. Of course, to be of use, the whole process should be as automatic as possible; the goal is to implement translators from χ into the input languages of several model checkers.

In [16], the translation of χ specifications into PROMELA is discussed in more generality. The translation proceeds in two phases. The first phase, which we will henceforth call the *preprocessing phase*, consists of a transformation of the χ model in an attempt to eliminate all χ constructs that do not directly map to PROMELA constructs. For instance, χ has an explicit construct for parallel composition which facilitates nested parallelism, whereas PROMELA only allows the (implicit) parallel composition of sequential PROMELA processes; so in the preprocessing phase the nested parallelism in the χ model is eliminated. If the result after the preprocessing phase is a χ model that only has constructions with a direct translation into PROMELA, then it can be translated to a PROMELA model; this phase is called the translation phase.

The main difficulty for establishing the correctness of the whole translation is that usually the two languages do not have a formal semantics in common. An advantage of the two-phase approach sketched above then is that the preprocessing phase of the translation, which is usually the most involved part, takes place entirely within the realm of χ . Therefore, a correctness proof for this phase only involves the formal semantics of χ . An additional advantage of the two-phase approach is that the preprocessing phase (and its correctness proof) is potentially reusable, e.g., when defining a translation from χ to some other language.

The appropriate correctness criterion for a translation depends of course on the application. If the purpose is to establish that a χ model is deadlock-free, then the translation should preserve deadlock. If the purpose is to do LTL model checking, then the translation should preserve the validity of LTL formulas. If the purpose is to do CTL model checking, then the translation should preserve the validity of all CTL formulas. In all cases, establishing the desired preservation of properties directly is usually cumbersome. It is often more convenient to relate the χ model and its transformation by establishing that they are related according to some behavioural equivalence pertaining to the operational semantics of χ . The purpose of this paper is to define a behavioural equivalence that can be used to establish the correctness of the preprocessing phase of translations of χ models into the language of state-based model checkers such as, e.g., SPIN or UPPAAL.

Of course, such a behavioural equivalence should then preserve the relevant properties, which in our case means that it should preserve deadlock and the validity of (state-based) CTL^*_{χ} formulas. Its intended application for establishing the correctness of syntactic translations puts some further requirements on the notion. For instance, since the transformations are generally defined in a compositional manner, it is particularly convenient if the behavioural equivalence is a congruence with respect to the syntactic constructs of χ . Also, it is desirable that it is defined on the operational semantics of χ as directly as possible, i.e., it should be bisimulation-like. We propose a notion of stuttering congruence that meets these requirements.

The paper is organised as follows.

In Section 2 we present the syntax and the operational semantics of the discrete-event and untimed part of the language χ . We use the operational semantics to

define when a χ -process has deadlock, and we give the semantics of CTL^*_{-X} formulas with respect to χ processes.

In Section 3, we propose an adaptation of divergence blind stuttering bisimilarity [14]. We add to it a termination condition, which takes care of the distinction between successful and unsuccessful termination present in χ , and a divergence condition, which is needed both for the preservation of deadlock and preservation of CTL^*_{-X} . We prove that our version of stuttering bisimilarity is an equivalence relation, and that it indeed preserves deadlock and the validity of (state-based) CTL^*_{-X} formulas.

In Section 4 we argue that stuttering bisimilarity as defined in Section 3 is not a congruence. So we adapt it further by excluding send and receive transitions as stuttering steps and by adding a root condition. The resulting notion we call stuttering congruence and we prove that it is indeed a congruence with respect to the syntactic constructs of the discrete-event, untimed part of χ .

In Section 5 we briefly discuss how our notion of stuttering congruence can be used to establish part of the correctness of the translation proposed in [16]. The paper ends with a conclusion.

2 The language χ

In this section we present the syntax and operational semantics of χ . We also define the notion of deadlock and the semantics of CTL^*_{-X} for χ processes. We use the formalization of χ proposed in [4], but without the time support and with a few minor differences that we shall mention on the fly.

2.1 Syntax and semantics

There are several predefined data types in χ , but they are not relevant for the present paper. For our purposes, it is enough to presuppose a set of variables V , a set of data values D , a set of data expressions E that includes D and V , and a set of boolean expressions B that includes the set of truth values $\{true, false\}$.

Definition 1. *A partial mapping $\sigma : V \rightarrow D$ with a finite domain (denoted $\text{dom}(\sigma)$) is called a state. The set of all states is denoted Σ .*

We assume that σ also extends to data expressions ($\sigma : E \rightarrow D$) and to boolean expressions ($\sigma : B \rightarrow \{true, false\}$). In the latter case we require σ to be total. To correctly override global variables by local ones of the same name, we use the function γ defined as:

$$\begin{aligned} \text{dom}(\gamma(\sigma_1, \sigma_2)) &= \text{dom}(\sigma_1) \cup \text{dom}(\sigma_2) \\ \gamma(\sigma_1, \sigma_2)(x) &= \begin{cases} \sigma_1(x), & \text{if } x \in \text{dom}(\sigma_1) \\ \sigma_2(x), & \text{if } x \in \text{dom}(\sigma_2) \setminus \text{dom}(\sigma_1). \end{cases} \end{aligned}$$

We now give the syntax of χ . The set of atomic processes A , and the set of all χ process terms P , are generated by the following grammar:

$$\begin{aligned} a &::= \varepsilon \mid \delta \mid \text{skip} \mid x := e \mid m!e \mid m?x \\ p &::= a \mid b \rightarrow p \mid p ; p \mid p \parallel p \mid p^* \mid p \parallel p \mid \llbracket s \mid p \rrbracket \mid \partial(p) . \end{aligned}$$

Here $a \in A$, $p \in P$, $x \in V$, $e \in E$, $b \in B$, $s \in \Sigma$ and $m \in M$, where M is a set of channel names.

Elements of the set $C = P \times \Sigma$ we call *configurations*. They represent processes together with their context. If $c = \langle p, \sigma \rangle$, then σ is the *state* of c . The semantics of χ is given in terms of configurations.

We make a distinction between successful and unsuccessful termination. The statement $c \downarrow$ denotes that $c \in C$ successfully terminates. The statement $c \xrightarrow{a} c'$ denotes that $c \in C$ can execute the action a and transform into a configuration c' . The set of actions that can be performed (denoted $\mathcal{A}$) consists of the internal action τ , the assignment action $aa(x, d)$, the send action $sa(m, d)$, the receive action $ra(m, d)$ and the communication action $ca(m, d)$, where $x \in V$, $m \in M$ and $d \in D$.

Atomic processes We explain each atomic process informally; the operational rules are given in Table 1.

The constant δ stands for the deadlock process. It cannot execute an action nor terminate successfully. The empty process ε cannot do an action either, but it is considered successfully terminated. The *skip* process performs the internal action τ (and terminates successfully). The assignment process $x := e$ assigns to x the value of the expression e according to the current state. The send process $m!e$ outputs the value of e (in the current state) along channel m . The receive process $m?x$ inputs a value along channel m and assigns it to x .

Table 1. Operational semantics for atomic processes

$\frac{}{\langle \varepsilon, \sigma \rangle \downarrow} \quad 1$	$\frac{}{\langle \text{skip}, \sigma \rangle \xrightarrow{\tau} \langle \varepsilon, \sigma \rangle} \quad 2$	$\frac{\sigma(e) = d}{\langle x := e, \sigma \rangle \xrightarrow{aa(x, d)} \langle \varepsilon, \gamma(\{x \mapsto d\}, \sigma) \rangle} \quad 3$
$\frac{\sigma(e) = d}{\langle m!e, \sigma \rangle \xrightarrow{sa(m, d)} \langle \varepsilon, \sigma \rangle} \quad 4$	$\frac{}{\langle m?x, \sigma \rangle \xrightarrow{ra(m, d)} \langle \varepsilon, \gamma(\{x \mapsto d\}, \sigma) \rangle} \quad 5$	

Compound processes Here we give an informal explanation for each of the seven operators; the operational rules are given in Table 2.

The guarded process $b \rightarrow p$ behaves as p when the value of the guard $b \in B$ is *true* (in the current state). The sequential composition $p ; q$ behaves as p followed by the process q . The alternative composition $p \parallel q$ stands for a non-deterministic choice between p and q . The process p^* behaves as p , executed

zero (successful termination), or more times. The parallel composition operator $\parallel$ executes p and q concurrently in an interleaved fashion. In addition, if one of the processes can execute a send action and the other one can execute a receive action on the same channel, then they can also communicate, i.e. $p \parallel q$ can also execute the communication action on this channel. The scope operator is used for declarations of local variables. The process $\llbracket s \mid p \rrbracket$ behaves as p in a local state s . In contrast to [4] and [15], channel declarations are not allowed, i.e. channels are global. Finally, the encapsulation operator ∂ disables all send and receive actions of a process. This is slightly more restrictive than in [4] and [15] where ∂ is parameterized by a set of actions that should be blocked, but corresponds to current practice.

Table 2. Operational semantics for composed processes

$\frac{\sigma(b) = \text{true}, \langle p, \sigma \rangle \downarrow}{\langle b := p, \sigma \rangle \downarrow} 6$	$\frac{\sigma(b) = \text{true}, \langle p, \sigma \rangle \xrightarrow{a} \langle p', \sigma' \rangle}{\langle b := p, \sigma \rangle \xrightarrow{a} \langle p', \sigma' \rangle} 7$	
$\frac{\langle p, \sigma \rangle \downarrow, \langle q, \sigma \rangle \downarrow}{\langle p ; q, \sigma \rangle \downarrow} 8$	$\frac{\langle p, \sigma \rangle \downarrow, \langle q, \sigma \rangle \xrightarrow{a} \langle q', \sigma' \rangle}{\langle p ; q, \sigma \rangle \xrightarrow{a} \langle q', \sigma' \rangle} 9$	$\frac{\langle p, \sigma \rangle \xrightarrow{a} \langle p', \sigma' \rangle}{\langle p ; q, \sigma \rangle \xrightarrow{a} \langle p' ; q, \sigma' \rangle} 10$
$\frac{\langle p, \sigma \rangle \downarrow}{\langle p \parallel q, \sigma \rangle \downarrow, \langle q \parallel p, \sigma \rangle \downarrow} 11$	$\frac{\langle p, \sigma \rangle \xrightarrow{a} \langle p', \sigma' \rangle}{\langle p \parallel q, \sigma \rangle \xrightarrow{a} \langle p', \sigma' \rangle, \langle q \parallel p, \sigma \rangle \xrightarrow{a} \langle p', \sigma' \rangle} 12$	
$\frac{}{\langle p^*, \sigma \rangle \downarrow} 13$	$\frac{\langle p, \sigma \rangle \xrightarrow{a} \langle p', \sigma' \rangle}{\langle p^*, \sigma \rangle \xrightarrow{a} \langle p' ; p^*, \sigma' \rangle} 14$	
$\frac{\langle p, \sigma \rangle \downarrow, \langle q, \sigma \rangle \downarrow}{\langle p \parallel q, \sigma \rangle \downarrow, \langle q \parallel p, \sigma \rangle \downarrow} 15$	$\frac{\langle p, \sigma \rangle \xrightarrow{a} \langle p', \sigma' \rangle}{\langle p \parallel q, \sigma \rangle \xrightarrow{a} \langle p' \parallel q, \sigma' \rangle, \langle q \parallel p, \sigma \rangle \xrightarrow{a} \langle q' \parallel p, \sigma' \rangle} 16$	
$\frac{\langle p, \sigma \rangle \xrightarrow{sa(m,d)} \langle p', \sigma \rangle, \langle q, \sigma \rangle \xrightarrow{ra(m,d)} \langle q', \sigma' \rangle}{\langle p \parallel q, \sigma \rangle \xrightarrow{ca(m,d)} \langle p' \parallel q', \sigma' \rangle, \langle q \parallel p, \sigma \rangle \xrightarrow{ca(m,d)} \langle q' \parallel p', \sigma' \rangle} 17$		
$\frac{\langle p, \gamma(s, \sigma) \rangle \downarrow}{\langle \llbracket s \mid p \rrbracket, \sigma \rangle \downarrow} 18$	$\frac{\langle p, \gamma(s, \sigma) \rangle \xrightarrow{a} \langle p', \sigma' \rangle}{\langle \llbracket s \mid p \rrbracket, \sigma \rangle \xrightarrow{a} \langle \llbracket \sigma' \upharpoonright \text{dom}(s) \mid p' \rrbracket, \gamma(\sigma, \sigma' \upharpoonright \text{dom}(\sigma) \setminus \text{dom}(s)) \rangle} 19$	
$\frac{\langle p, \sigma \rangle \downarrow}{\langle \partial(p), \sigma \rangle \downarrow} 20$	$\frac{\langle p, \sigma \rangle \xrightarrow{a} \langle p', \sigma' \rangle, a \notin \{sa(m,d), ra(m,d)\}}{\langle \partial(p), \sigma \rangle \xrightarrow{a} \langle \partial(p'), \sigma' \rangle} 21$	

2.2 Deadlock and $\text{CTL}^*_{-\chi}$ in χ

First we give some abbreviations: $c \rightarrow c'$ denotes that there exists $a \in A$ such that $c \xrightarrow{a} c'$; $c \not\rightarrow$ denotes that there does not exist c' such that $c \rightarrow c'$. Now we define when a configuration has a deadlock.

Definition 2. We say that $c \in C$ has a deadlock iff there exist $c_0, \dots, c_n \in C$ such that

$$c_0 = c, c_0 \rightarrow \dots \rightarrow c_n, c_n \not\rightarrow \text{ and } c_n \not\downarrow.$$

Next, we recall the formulas of the logic CTL^*_X [7] and give their semantics. Let AP be a set that we call the set of atomic propositions.

Definition 3. The formulas of the logic CTL^*_X are defined as follows:

1. every atomic proposition is a state formula;
2. if φ is a state formula, then $\neg\varphi$ is a state formula;
3. if φ_1 and φ_2 are state formulas, then $\varphi_1 \wedge \varphi_2$ is a state formula;
4. if ψ is a path formula, then $\exists\psi$ is a state formula;
5. if φ is a state formula, then φ is a path formula;
6. if ψ is a path formula, then $\neg\psi$ is a path formula;
7. if ψ_1 and ψ_2 are path formulas, then $\psi_1 \wedge \psi_2$ is a path formula;
8. if ψ_1 and ψ_2 are path formulas, then $\psi_1 \mathcal{U} \psi_2$ is a path formula.

For the satisfaction relation we need the notion of a path.

Definition 4. A path is an infinite sequence of configurations $c_0, c_1, c_2, \dots$ such that either:

- a. $c_0 \rightarrow c_1 \rightarrow c_2 \rightarrow \dots$ or
- b. $c_0 \rightarrow \dots \rightarrow c_n, c_n \not\rightarrow$ and $c_{i+1} = c_i$ for all $i \geq n$.

If π is a path $c_0, c_1, c_2, \dots$ then π^i denotes the path $c_i, c_{i+1}, c_{i+2}, \dots$. Sometimes we call π a path from c_0 .

We require that two configurations with the same state satisfy the same atomic propositions, by assuming that for each state $\sigma \in \Sigma$ there is a mapping $val_\sigma : AP \rightarrow \{true, false\}$. It will be used in the definition of the satisfaction relation between configurations and CTL^*_X formulas.

Definition 5. We simultaneously define the satisfaction of a state formula φ by a configuration c (notation: $c \models \varphi$) and the satisfaction of a path formula ψ by a path π (notation: $\pi \models \psi$) as follows:

1. $c \models \alpha \in AP$ iff $val_c(\alpha) = true$
2. $c \models \neg\varphi$ iff $c \not\models \varphi$,
3. $c \models \varphi_1 \wedge \varphi_2$ iff $c \models \varphi_1$ and $c \models \varphi_2$,
4. $\pi \models \varphi$ iff $c \models \varphi$ where c is the first configuration in path π ,
5. $c \models \exists\psi$ iff there is a path π from c such that $\pi \models \psi$,
6. $\pi \models \neg\psi$ iff $\pi \not\models \psi$,
7. $\pi \models \psi_1 \wedge \psi_2$ iff $\pi \models \psi_1$ and $\pi \models \psi_2$,
8. $\pi \models \psi_1 \mathcal{U} \psi_2$ iff there exists $j \geq 0$ such that $\pi^j \models \psi_2$ and $\pi^i \models \psi_1$ for all $i < j$.

3 Stuttering bisimilarity

Stuttering equivalence was originally proposed and proved to preserve the validity of CTL^*_X formulas by Browne, Clarke and Grumberg [5]. They define the notion on maximal paths associated with total Kripke structures, i.e., Kripke structures without deadlocked states. De Nicola and Vaandrager [14] drop the requirement that Kripke structures are total, and provide a definition of stuttering equivalence that proceeds via a notion of divergence blind stuttering bisimilarity defined on the Kripke structures themselves. We take divergence blind stuttering bisimilarity as a starting point, and add to it two conditions:

1. a *termination condition* that ensures a proper handling of the distinction between successful and unsuccessful termination as it is present in χ ; and
2. a *divergence condition* similar to one that appears in [10] to ensure the preservation of deadlock and the preservation of CTL^*_X .

Remark 1. To obtain a notion that coincides with the notion of [5], instead of adding a divergence condition, de Nicola and Vaandrager define a divergence sensitive version of stuttering bisimilarity by extending Kripke structures with a fresh state that serves as a sink-state for deadlocked or divergent states. This approach is not suitable in our case, because it identifies deadlock and livelock, and because it is in conflict with our requirement, mentioned in the introduction, that the equivalence is defined directly on the operational semantics of χ .

Definition 6. A symmetric relation $R \subseteq C \times C$ is a stuttering bisimulation iff, for all $(c, d) \in R$, c and d have the same state and:

1. if $c \downarrow$, then there exist $d_0, \dots, d_n \in C$ such that

$$d_0 = d, d_0 \rightarrow \dots \rightarrow d_n, d_n \downarrow \text{ and } c R d_i \text{ for all } i \leq n,$$

2. if $c \rightarrow c'$ for some $c' \in C$, then there exist $d_0, \dots, d_n \in C$ such that

$$d_0 = d, d_0 \rightarrow \dots \rightarrow d_n, c R d_i \text{ for all } i \leq n-1, \text{ and } c' R d_n,$$

3. if there exists an infinite sequence $c_0, c_1, c_2, \dots \in C$ such that

$$c_0 = c, c_0 \rightarrow c_1 \rightarrow c_2 \rightarrow \dots \text{ and } c_i R d \text{ for all } i \geq 0,$$

then there exist $d' \in C$ and $j > 0$ such that

$$d \rightarrow d' \text{ and } c_j R d'.$$

We refer to condition 1 as the termination condition, to condition 2 as the transfer condition, and to condition 3 as the divergence condition.

A non-empty and finite sequence of configurations we call a *block*. If $B = c_0, \dots, c_n$ and $C = d_0, \dots, d_m$ are blocks and R is a stuttering bisimulation, we write $B R C$ when $c_0 R d_0$, $c_n R d_m$ and when, for all $i < n, j < m$, $c_i R d_j$ implies $c_{i+1} R d_j$ or $c_i R d_{j+1}$. Note that, $B R C$ implies $C R B$.

Definition 7. Let R be a stuttering bisimulation. Two sequences of configurations Ω_1 and Ω_2 are R -corresponding if:

- they are both finite and can be partitioned as $\Omega_1 = B^0, \dots, B^K$ and $\Omega_2 = C^0, \dots, C^K$ where $B^k R C^k$ for all $0 \leq k \leq K$; or
- they are both infinite and can be partitioned as $\Omega_1 = B^0, B^1, B^2 \dots$ and $\Omega_2 = C^0, C^1, C^2, \dots$ where $B^k R C^k$ for all $k \geq 0$.

It is clear that R -correspondence is a symmetric relation. Also note that, if $c_0, \dots, c_n$ and $d_0, \dots, d_m$ (similarly for the infinite case) are R -corresponding then for all $i \leq n$ there exists $j \leq m$ and for all $j \leq m$ there exists $i \leq n$ such that $c_i R d_j$.

We now prove some properties of a stuttering bisimulation R .

Lemma 1. Suppose $c R d$, $c_0, \dots, c_n \in C$, $c_0 = c$ and $c_0 \rightarrow \dots \rightarrow c_n$. Then there exist $d_0, \dots, d_m \in C$, such that $d_0 = d$, $d_0 \rightarrow \dots \rightarrow d_m$ and that $c_0, \dots, c_n$ and $d_0, \dots, d_m$ are R -corresponding.

Proof. The proof goes by induction on n . If $n = 0$, we take $m = 0$. Assume that the statement is valid for all $n \geq 0$ and suppose $c_0 \rightarrow \dots \rightarrow c_n \rightarrow c_{n+1}$. From the inductive hypothesis, there exist $d_0, \dots, d_m \in C$ such that $d_0 = d$ and $d_0 \rightarrow \dots \rightarrow d_m$, and $c_0, \dots, c_n$ and $d_0, \dots, d_m$ can be partitioned as $B^0, \dots, B^K$ and $C^0, \dots, C^K$ respectively, such that $B^k R C^k$ for all $k \leq K$. This implies $c_n R d_m$. Since $c_n \rightarrow c_{n+1}$, there exist $d_m^0, \dots, d_m^k$ such that

$$d_m^0 = d_m, d_m^0 \rightarrow \dots \rightarrow d_m^k, c_n R d_m^j \text{ for all } j \leq k-1, \text{ and } c_{n+1} R d_m^k.$$

Lemma 2. If $c R d$, then

- a. for every sequence of configurations $c_0, \dots, c_n$ such that $c_0 = c$, $c_0 \rightarrow \dots \rightarrow c_n$ and $c_n \downarrow$ there exists an R -corresponding sequence $d_0, \dots, d_m$ such that

$$d_0 = d, d_0 \rightarrow \dots \rightarrow d_m \text{ and } d_m \downarrow; \text{ and}$$

- b. for every sequence of configurations $c_0, \dots, c_n$ such that $c_0 = c$, $c_0 \rightarrow \dots \rightarrow c_n$ and $c_n \nrightarrow$ there exists an R -corresponding sequence $d_0, \dots, d_m$ such that

$$d_0 = d, d_0 \rightarrow \dots \rightarrow d_m, \text{ and } d_m \nrightarrow.$$

Proof. Consider a sequence of configurations $c_0, \dots, c_n$ such that $c_0 = c$ and $c_0 \rightarrow \dots \rightarrow c_n$. Then, by Lemma 1, there exists an R -corresponding sequence $d_0, \dots, d_m$ such that $d_0 = d$ and $d_0 \rightarrow \dots \rightarrow d_m$; it follows that $c_n R d_m$.

- a. Suppose $c_n \downarrow$. Then, there exist $d_m^0, \dots, d_m^k \in C$ such that

$$d_m^0 = d_m, d_m^0 \rightarrow \dots \rightarrow d_m^k, d_m^k \downarrow \text{ and } c_n R d_m^j \text{ for all } 0 \leq j \leq k.$$

- b. Suppose $c_n \not\rightarrow$. There cannot exist an infinite sequence of configurations $d_m^0, d_m^1, d_m^2, \dots$ such that

$$d_m^0 = d_m, \quad d_m^0 \rightarrow d_m^1 \rightarrow d_m^2 \rightarrow \dots \quad \text{and} \quad c_n R d_m^i \quad \text{for all } i \geq 0,$$

for then $c_n \not\rightarrow$ would contradict the divergence condition. Also, there cannot exist $d_m^0, \dots, d_m^k \in C$ ($k \geq 0$) such that

$$d_m^0 = d_m, \quad d_m^0 \rightarrow \dots \rightarrow d_m^k, \quad c_n R d_m^i \quad \text{for all } i \leq k-1, \quad \text{and not } c_n R d_m^k,$$

for then $c_n \not\rightarrow$ would contradict the transfer condition. Hence, there exist $d_m^0, \dots, d_m^k \in C$ such that

$$d_m^0 = d_m, \quad d_m^0 \rightarrow \dots \rightarrow d_m^k, \quad d_m^k \not\rightarrow, \quad \text{and } c_n R d_m^i \quad \text{for all } i \leq k.$$

Clearly, in both cases $c_0, \dots, c_n$ and $d_0, \dots, d_m, d_m^1, \dots, d_m^k$ are R -corresponding.

Lemma 3. *If $c R d$, then for every infinite sequence of configurations $c_0, c_1, c_2, \dots$ such that $c_0 = c$ and $c_0 \rightarrow c_1 \rightarrow c_2 \rightarrow \dots$, there exists an R -corresponding sequence of configurations $d_0, d_1, d_2, \dots$ such that*

$$d_0 = d, \quad d_0 \rightarrow d_1 \rightarrow d_2 \rightarrow \dots$$

Proof. Consider an infinite sequence of configurations $c_0, c_1, c_2, \dots$ such that $c_0 = c$ and $c_0 \rightarrow c_1 \rightarrow c_2 \rightarrow \dots$; we define infinite sequences of blocks $C^0, C^1, C^2, \dots$ and $D^0, D^1, D^2, \dots$ such that $C^k R D^k$ for all $k \geq 0$, and such that $C^0, C^1, C^2, \dots$ is a partitioning of $c_0, c_1, c_2, \dots$ and $D^0, D^1, D^2, \dots$ is a partitioning of a sequence $d_0, d_1, d_2, \dots$ such that $d_0 = d$ and $d_0 \rightarrow d_1 \rightarrow d_2 \rightarrow \dots$. The construction is by induction on k , and each step delivers the blocks C^k and D^k and the first configurations of C^{k+1} and D^{k+1} .

Let $C^0, \dots, C^K$ and $D^0, \dots, D^K$ be sequences of blocks such that $C^k R D^k$ for all $k \geq 0$, and let c_n and d_m be the first configurations of C^{k+1} and D^{k+1} respectively.

If $c_i R d_m$ for all $i \geq n$, then by the divergence condition there exist $d'_m \in C$ and $j > n$ such that $d_m \rightarrow d'_m$ and $c_j R d'_m$. Then, take

$$C^{K+1} = c_n, \dots, c_{j-1} \quad \text{and} \quad D^{K+1} = d_m,$$

and take c_j and d'_m for the first configurations of C^{k+2} and D^{k+2} respectively. On the other hand, if $c_i R d_m$ for all $i \leq k$ but not $c_{k+1} R d_m$, then, since $c_k R d_m$, there exist $d_m^0, \dots, d_m^l \in C$ such that

$$d_m^0 = d_m, \quad d_m^0 \rightarrow \dots \rightarrow d_m^l, \quad c_k R d_m^i \quad \text{for all } i \leq l-1, \quad \text{and } c_{k+1} R d_m^l.$$

Note that $l > 0$ because not $c_{k+1} R d_m$. Take

$$C^{k+1} = c_n, \dots, c_k \quad \text{and} \quad D^{k+1} = d_m^0, \dots, d_m^{l-1}.$$

Take c_{k+1} and d_m^l for the first configurations of C^{k+2} and D^{k+2} respectively.

Definition 8. Two configurations c and d are stuttering bisimilar, denoted $c \sim_{st} d$, if there exists a stuttering bisimulation R such that cRd .

We now prove that it is stuttering bisimilarity is an equivalence relation. The usual way to prove that a bisimulation-like equivalence $\sim$ is transitive, is to suppose that $c \sim d$ and $d \sim e$ are witnessed by bisimulation relations R_1 and R_2 respectively, and then show that $R_1 \circ R_2$ is again a bisimulation relation. However, this method fails here, due to the nature of the divergence condition. We prove transitivity by showing that the transitive closure of a stuttering bisimulation is a stuttering bisimulation.

Lemma 4. Let R be a stuttering bisimulation. Then $R^+ = \bigcup_{o \in \omega} R^o$ is also a stuttering bisimulation.

Proof. The transitive closure of any symmetric relation is symmetric so R^+ is symmetric.

Suppose $c R^+ e$. First we prove that, for all $o \geq 1$ and all $(c, e) \in R^o$, c and e have the same state, (c, e) satisfies the termination and transfer condition and if $c_0 = c$, $c_0 \rightarrow c_1 \rightarrow c_2 \rightarrow \dots$ then there exist $e_0, e_1, e_2, \dots \in C$ such that $e_0 = e$, $e_0 \rightarrow e_1 \rightarrow e_2 \rightarrow \dots$ and for all $j \geq 0$ there is $i \geq 0$ such that $c_i R^o e_j$.

The proof is by induction on o . If $o = 1$ then the statement follows from the definition of R and Lemma 3. Suppose the statement is valid for $o \geq 1$ and suppose $c R^{o+1} e$. This means that there is $d \in C$ such that $c R^o d$ and $d R e$. By the inductive hypothesis, c and d have the same state. By the definition of R , d and e have the same state. Therefore, c and e have the same state.

Suppose $c \downarrow$. By the inductive hypothesis, there exist $d_0, \dots, d_n \in C$ such that

$$d_0 = d, d_0 \rightarrow \dots \rightarrow d_n, d_n \downarrow \text{ and } c R^o d_i \text{ for all } i \leq n.$$

By Lemma 2a, there exist $e_0, \dots, e_m \in C$ such that

$$e_0 = e, e_0 \rightarrow \dots \rightarrow e_m, e_m \downarrow \text{ and for all } j \leq m \text{ there is } i \leq n \text{ such that } d_i R e_j$$

(see the note after Definition 7). Therefore, $c R^{o+1} e_j$ for all $j \leq m$.

Suppose $c \rightarrow c'$. By the inductive hypothesis, there exist $d_0, \dots, d_n \in C$ such that

$$d_0 = d, d_0 \rightarrow \dots \rightarrow d_n, c R^o d_i \text{ for all } i \leq n-1, \text{ and } c' R^o d_n.$$

By Lemma 1, there exist $e_0, \dots, e_m \in C$ such that $e_0 = e$,

$$e_0 \rightarrow \dots \rightarrow e_m, d_{n-1} R e_m, \text{ and for all } j \leq m \text{ there is } i \leq n-1 \text{ such that } d_i R e_j.$$

Now, there exist $e_m^0, \dots, e_m^k \in C$ such that

$$e_m^0 = e_m, e_m^0 \rightarrow \dots \rightarrow e_m^k, d_{n-1} R e_m^i \text{ for all } i \leq k-1, \text{ and } d_n R e_m^k.$$

Since $c R^o d_i$ and $d_i R e_j$, $c R^{o+1} e_j$ for all $j \leq m$. Since $c R^o d_{n-1}$ and $d_{n-1} R e_m^i$, $c R^{o+1} e_m^i$ for all $i \leq k-1$. Since $c' R^o d_n$ and $d_n R e_m^k$, $c' R^{o+1} e_m^k$.

Suppose now $c_0 \rightarrow c_1 \rightarrow c_2 \rightarrow \dots$. By the inductive hypothesis, there exists an infinite sequence of configurations $d_0, d_1, d_2, \dots \in C$ such that

$$d_0 = d, d_0 \rightarrow d_1 \rightarrow d_2 \rightarrow \dots \text{ and for all } j \geq 0 \text{ there is } i \geq 0 \text{ such that } c_i R^o d_j.$$

By Lemma 3 there exist $e_0, e_1, e_2, \dots \in C$ such that

$$e_0 = e, e_0 \rightarrow e_1 \rightarrow e_2 \rightarrow \dots \text{ and for all } k \geq 0 \text{ there is } j \geq 0 \text{ such that } d_j R e_k.$$

Therefore, for all $k \geq 0$ there is $j \geq 0$ such that $c_i R^{o+1} e_k$.

Since $R^+ = \bigcup_{o \in \omega} R^o$, it now follows immediately that R^+ satisfies the termination and transfer condition and that, for all $(c, d) \in R^+$, c and d have the same state. It remains to prove that R^+ also satisfies the divergence condition.

Suppose $c_0 \rightarrow c_1 \rightarrow c_2 \rightarrow \dots$ and $c_i R^+ d$ for all $i \geq 0$. Since $c R^+ d$, we conclude that $c R^o d$ for some $o \geq 1$. As we showed before, there exist $d_0, d_1, d_2, \dots \in C$ such that $d_0 = d, d_0 \rightarrow d_1 \rightarrow d_2 \rightarrow \dots$ and for all $j \geq 0$ there exists $i \geq 0$ such that $c_i R^o d_j$. In particular, there exists $i \geq 0$ such that $c_i R^+ d_1$. If $i > 0$ then the divergence condition is proved, so suppose $i = 0$. Then $c_1 R^+ d_0, c_0 R^+ d_0$ and $c_0 R^+ d_1$, and hence, since R^+ is symmetric and transitive, it follows that $c_1 R^+ d_1$.

Theorem 1. *Stuttering bisimilarity on configurations is an equivalence relation.*

Proof. The set $\{(c, c) \mid c \in C\}$ is clearly a stuttering bisimulation, so $\sim_{st}$ is reflexive. Furthermore, that $\sim_{st}$ is symmetric follows directly from the required symmetry of a stuttering bisimulation. It remains to prove transitivity.

Suppose $c \sim_{st} d$ and $d \sim_{st} e$. Then, there exist stuttering bisimulations R_1 and R_2 such that $c R_1 d$ and $d R_2 e$. Let $R = R_1 \cup R_2$. It is not hard to show that R is also a stuttering bisimulation. By Lemma 4, so is R^+ . Since $R \subseteq R^+$, $c R^+ d$ and $d R^+ e$. By the transitivity of R^+ , we conclude $c R^+ e$, and hence $c \sim_{st} e$.

Corollary 1. *$\sim_{st}$ -correspondence is an equivalence relation.*

In the remainder of this section we establish that stuttering bisimilarity preserves deadlock and the validity of CTL^*_{-X} formulas.

Theorem 2. *If $c \sim_{st} d$ then c has deadlock iff d has deadlock.*

Proof. Suppose c has deadlock (when d has deadlock the proof is similar). This means that there exist $c_0, \dots, c_n$ such that

$$c_0 = c, c_0 \rightarrow \dots \rightarrow c_n, c_n \not\rightarrow \text{ and } c_n \not\downarrow.$$

By Lemma 2b, there exist $d_0, \dots, d_m$ such that

$$d_0 = d, d_0 \rightarrow \dots \rightarrow d_m, d_m \not\rightarrow \text{ and } c_n R d_m.$$

Suppose $d_m \downarrow$. Then, there exist $c_n^0, \dots, c_n^k$ such that

$$c_n^0 \rightarrow \dots \rightarrow c_n^k, \text{ and } c_n^k \downarrow.$$

This is however not possible (even when $k = 0$) because $c_n \not\rightarrow$ and $c_n \not\downarrow$.

The following lemma plays a crucial role in the proof that stuttering bisimilarity preserves the validity of CTL^*_{-X} formulas.

Lemma 5. *If $c \sim_{st} d$, then for every path from c there is a $\sim_{st}$ -corresponding path from d .*

Proof. Let $c_0, c_1, c_2, \dots$ be a path from c . There are two cases:

- if $c_0 \rightarrow \dots \rightarrow c_n, c_n \not\rightarrow$ and $c_{i+1} = c_i$ for all $i \geq n$, then the statement follows directly from Lemma 2b;
- if $c_0 \rightarrow c_1 \rightarrow c_2 \rightarrow \dots$, then the statement follows directly from Lemma 3.

Now we present the main theorem.

Theorem 3. *If $c \sim_{st} d$ then for all CTL^*_{-X} formulas φ , $c \models \varphi$ iff $d \models \varphi$.*

Proof. The proof is a slight adaptation of the proof of Theorem 3.2.3 from [14]. Let c and d be two configurations such that $c \sim_{st} d$. Let π_p and π_q be two stuttering equivalent paths from c and d respectively. Let φ be a CTL^*_{-X} formula. We prove:

1. if φ is a state formula, then $c \models \varphi$ iff $d \models \varphi$, and
2. if φ is a path formula, then $\pi_p \models \varphi$ iff $\pi_q \models \varphi$

The proof goes by the structural induction on φ , first for state and then for path formulas.

1. Suppose $\varphi = \alpha$ for $\alpha \in AP$. We have $c \models \alpha$ iff $val_\sigma(\alpha) = true$ iff $d \models \alpha$.
2. Suppose $\varphi = \neg\varphi'$. We have $c \models \varphi$ iff $c \not\models \varphi'$. From the inductive hypothesis this is equivalent to $d \not\models \varphi'$, which in turn is equivalent to $d \models \varphi$.
3. Suppose $\varphi = \varphi_1 \wedge \varphi_2$. We have $c \models \varphi$ iff $c \models \varphi_1$ and $c \models \varphi_2$. From the inductive hypothesis this is equivalent to $d \models \varphi_1$ and $d \models \varphi_2$, which is in turn equivalent to $d \models \varphi$.
4. Suppose $\varphi = \exists\psi$. Let $c \models \varphi$. This implies that there is a path π'_p from p such that $\pi'_p \models \psi$. Let π'_q be a path from q such that $\pi'_p \sim_{st} \pi'_q$ (from Lemma 5, π'_q always exists). From the inductive hypothesis we obtain $\pi_q \models \psi$, which implies $d \models \varphi$.
Symmetrically when $d \models \varphi$.
5. Suppose φ is a path formula. We have $\pi_p \models \varphi$ iff $c \models \varphi$. From the inductive hypothesis $c \models \varphi$ is equivalent to $d \models \varphi$, which is in turn equivalent to $\pi_q \models \varphi$.
6. Suppose $\varphi = \neg\psi'$. We have $\pi_p \models \varphi$ iff $\pi_p \not\models \psi'$. From the inductive hypothesis $\pi_p \not\models \psi'$ is equivalent to $\pi_q \not\models \psi'$, which is in turn equivalent to $\pi_q \models \varphi$.
7. Suppose $\varphi = \psi_1 \wedge \psi_2$. We have $\pi_p \models \varphi$ iff $\pi_p \models \psi_1$ and $\pi_p \models \psi_2$. From the inductive hypothesis this is equivalent to $\pi_q \models \psi_1$ and $\pi_q \models \psi_2$, which is in turn equivalent to $\pi_q \models \varphi$.

8. Suppose $\varphi = \psi_1 \mathcal{U} \psi_2$. Let $\pi_p \models \varphi$. This implies that there exists $j_p \geq 0$ such that

$$\pi_p^{j_p} \models \psi_2 \text{ and } \pi_p^{i_p} \models \psi_1 \text{ for all } 0 \leq i_p < j_p.$$

Let π_p and π_q be partitioned as

$$\pi_p = B^0, B^1, B^2, \dots \text{ and } \pi_q = C^0, C^1, C^2, \dots$$

such that for all $k \geq 0$, $B^k \sim_{st} C^k$. Let B^K be the block that contains the first configuration of $\pi_p^{j_p}$. It is clear that

$$\pi_p^{j_p} \sim_{st} B^K, B^{K+1}, B^{K+2}, \dots \sim_{st} C^K, C^{K+1}, C^{K+2}, \dots$$

From $\pi_p^{j_p} \models \psi_2$ and the inductive hypothesis we conclude that

$$C^K, C^{K+1}, C^{K+2}, \dots \models \psi_2.$$

Let $j_q \geq 0$ be such that $\pi_q^{j_q} = C^K, C^{K+1}, \dots$ and let $i_q < j_q$ be an arbitrary index. Let C^L be the block that contains the first configuration of $\pi_q^{j_q}$. We have

$$\pi_q^{i_q} \sim_{st} C^L, C^{L+1}, C^{L+2}, \dots \sim_{st} B^L, B^{L+1}, B^{L+2}, \dots$$

From $i_q < j_q$ we have $L < K$. This implies

$$B^L, B^{L+1}, B^{L+2}, \dots \models \psi_1.$$

By the inductive hypothesis, $\pi_q^{i_q} \models \psi_1$.

The proof for the other direction is symmetric.

4 Stuttering congruence

First we extend the definition of stuttering bisimilarity to the level of χ processes.

Definition 9. *Two processes p and q are stuttering bisimilar, denoted $p \sim_{st} q$, if for all $\sigma \in \Sigma$, $\langle p, \sigma \rangle \sim_{st} \langle q, \sigma \rangle$.*

To see that stuttering bisimilarity is not a congruence on χ processes, consider the following example.

Example 1. Note that the execution of a send action does not affect the state (see Rule 4 in Table 1), so $a!0 \sim_{st} b!0$. However, $a!0 \parallel a?x \not\sim_{st} b!0 \parallel a?x$, for the process on the left can do a communication action and change the value of x , whereas the process on the right cannot. It follows that $\sim_{st}$ is not a congruence for parallel composition. Also note that $a!0 \sim_{st} skip$, whereas $\partial(a!0) \not\sim_{st} \partial(skip)$ ($\partial(a!0)$ is deadlocked; $\partial(skip)$ does a τ -transition and terminates successfully). So $\sim_{st}$ is not a congruence for encapsulation either.

The example shows that for an equivalence on χ processes to be a congruence, it should not be completely action insensitive. We adapt the definition of stuttering bisimilarity in such a way that it distinguishes the send and receive actions from the other actions.

Let $\mathcal{A}^{com}$ be the set of all send and receive actions. Let $c \hookrightarrow c'$ mean that there is an $a \in \mathcal{A} \setminus \mathcal{A}^{com}$ such that $c \xrightarrow{a} c'$. Furthermore, let $c \xrightarrow{(a)} c'$ denote $c \xrightarrow{a} c'$ when $a \in \mathcal{A}^{com}$, and $c \hookrightarrow c'$ when $a \in \mathcal{A} \setminus \mathcal{A}^{com}$.

Definition 10. A symmetric relation $R \subseteq C \times C$ is an interaction sensitive stuttering bisimulation iff, for all $(c, d) \in R$, c and d have the same state and:

1. if $c \downarrow$, then there exist $d_0, \dots, d_n \in C$ such that

$$d_0 = d, d_0 \hookrightarrow \dots \hookrightarrow d_n, d_n \downarrow \text{ and } c R d_i \text{ for all } i \leq n,$$

2. if $c \xrightarrow{a} c'$, then there exist $d_0, \dots, d_n \in C$ such that

$$d_0 = d, d_0 \hookrightarrow \dots \hookrightarrow d_{n-1} \xrightarrow{(a)} d_n, c R d_i \text{ for all } i \leq n-1, \text{ and } c' R d_n$$

(we allow n to be 0 only if $a \in \mathcal{A} \setminus \mathcal{A}^{com}$ and c and c' have the same state),

3. if there exists an infinite sequence $c_0, c_1, c_2, \dots \in C$ such that

$$c_0 = c, c_0 \hookrightarrow c_1 \hookrightarrow c_2 \hookrightarrow \dots \text{ and } c_i R d \text{ for all } i \geq 0,$$

then there exist $d' \in C$ and $j > 0$ such that $d \hookrightarrow d'$ and $c_j R d'$.

Two configurations c and d are interaction sensitive stuttering bisimilar, denoted $c \sim_{isst} d$, if there exists an interaction sensitive stuttering bisimulation R such that $c R d$.

Theorem 4. Interaction sensitive stuttering bisimilarity is an equivalence.

Proof. Reflexivity and symmetry are proved as before. For the transitivity proof, it can be easily seen that Lemmas 1, 2a, and 3 hold when $\rightarrow$ is replaced by $\hookrightarrow$. Then, the proof goes similarly as for Theorem 1.

To show that all the previous results hold, we prove the following theorem.

Theorem 5. Interaction sensitive stuttering bisimilarity is a stuttering bisimulation.

Proof. Suppose that $c \sim_{isst} d$ for some $c, d \in C$. To show that the termination and transfer condition hold is trivial since $c \hookrightarrow c'$ implies $c \rightarrow c'$ for all $c, c' \in C$. We verify the divergence condition in detail.

Suppose

$$c_0 \rightarrow c_1 \rightarrow c_2 \rightarrow \dots \text{ and } c_i \sim_{isst} d \text{ for all } i \geq 0.$$

If every arrow in this sequence is a ' $\hookrightarrow$ ' arrow, we have that there exist d' and $i > 0$ such that $d \hookrightarrow d'$ and $c_i \sim_{isst} d'$. In this case we are done because $d \hookrightarrow d'$ implies $d \rightarrow d'$.

Suppose now

$$c_0 \hookrightarrow \dots \hookrightarrow c_n \xrightarrow{a} c_{n+1} \text{ and } a \in \mathcal{A}^{com}.$$

Since $c_n \sim_{isst} d$, there exist $d_0, \dots, d_m$ such that $d_0 = d$,

$$d_0 \hookrightarrow \dots \hookrightarrow d_{m-1} \xrightarrow{a} d_m, c_n \sim_{isst} d_i \text{ for all } i \leq m-1 \text{ and } c_{n+1} \sim_{isst} d_m.$$

Because $a \in \mathcal{A}^{com}$, $m > 0$. Then, $c_n \sim_{isst} d_1$ or $c_{n+1} \sim_{isst} d_1$. If $c_n \sim_{isst} d_1$ and $n = 0$ then, since $c_0 \sim_{isst} d_0$, $c_1 \sim_{isst} d_0$ and $\sim_{isst}$ is symmetric and transitive, we conclude that $c_1 \sim_{isst} d_1$.

Interaction sensitive stuttering bisimulation is not a congruence for alternative composition.

Example 2. Note that $skip \sim_{isst} skip$ and $\delta \sim_{isst} skip ; \delta$. However, $skip \parallel \delta \not\sim_{isst} skip \parallel skip ; \delta$, for the right-hand side process can execute $skip$ and then deadlock, while the left-hand side process never deadlocks. So, interaction sensitive stuttering bisimulation is not a congruence for alternative composition.

Both the problem illustrated in the above example, and its solution are well known; we need to add a root condition.

Definition 11. Two configurations c and d are rooted interaction sensitive stuttering bisimilar (notation: $c \sim_{riss} d$) iff

1. $c \downarrow$ iff $d \downarrow$
2. if $c \xrightarrow{a} c'$, then there exists $d' \in C$ such that $d \xrightarrow{(a)} d'$ and $c' \sim_{isst} d'$,
3. if $d \xrightarrow{a} d'$, then there exists $c' \in C$ such that $c \xrightarrow{(a)} c'$ and $c' \sim_{isst} d'$,

It is clear that $c \sim_{riss} d$ implies $c \sim_{isst} d$. Also, that rooted interaction sensitive stuttering bisimilarity is an equivalence relation is easily proved. The notion induces a congruence on χ processes.

Definition 12. Two processes p and q are stuttering congruent, denoted $p \cong_{st} q$, if $\langle p, \sigma \rangle \sim_{riss} \langle q, \sigma \rangle$ for all $\sigma \in \Sigma$.

Clearly, $\cong_{st}$ is an equivalence relation and $p \cong_{st} q$ implies $p \sim_{st} q$, and it is a congruence for the constructs of χ .

Theorem 6. For all $p, q, \bar{p}, \bar{q} \in P$, if $p \cong_{st} q$ and $\bar{p} \cong_{st} \bar{q}$, then:

1. $b : \rightarrow p \cong_{st} b : \rightarrow q$,
2. $p ; \bar{p} \cong_{st} q ; \bar{q}$,
3. $p \parallel \bar{p} \cong_{st} q \parallel \bar{q}$,
4. $p^* \cong_{st} q^*$,
5. $p \parallel \bar{p} \cong_{st} q \parallel \bar{q}$,
6. $\llbracket s \mid p \rrbracket \cong_{st} \llbracket s \mid q \rrbracket$ for all states s ,
7. $p \parallel \bar{p} \cong_{st} q \parallel \bar{q}$,
8. $\partial(p) \cong_{st} \partial(q)$.

Proof. For each case we only check the conditions of rooted interaction sensitive stuttering bisimilarity in one direction since the other direction is symmetric.

1. To prove the first condition, suppose $\langle b : \rightarrow p, \sigma \rangle \downarrow$. Note that Rule 6 in Table 2 is the final rule of any derivation with $\langle b : \rightarrow p, \sigma \rangle \downarrow$ as conclusion, so it holds that $\sigma(b) = \text{true}$ and $\langle p, \sigma \rangle \downarrow$. Hence, since $\langle p, \sigma \rangle \cong_{st} \langle q, \sigma \rangle$, it follows that $\langle q, \sigma \rangle \downarrow$, so $\langle b : \rightarrow q, \sigma \rangle \downarrow$ by Rule 6. Similarly, from $\langle b : \rightarrow q, \sigma \rangle \downarrow$ one can prove that $\langle b : \rightarrow p, \sigma \rangle \downarrow$.

To prove the second condition, suppose $\langle b : \rightarrow p, \sigma \rangle \xrightarrow{a} \langle r, \sigma' \rangle$. Since Rule 7 must be the final rule of any derivation of this transition, it holds that $\sigma(b) = \text{true}$, $\langle p, \sigma \rangle \xrightarrow{a} \langle p', \sigma' \rangle$ and $r = p'$. Since $\langle p, \sigma \rangle \cong_{st} \langle q, \sigma \rangle$, there exists q' such that $\langle q, \sigma \rangle \xrightarrow{(a)} \langle q', \sigma' \rangle$ and $\langle p', \sigma' \rangle \sim_{isst} \langle q', \sigma' \rangle$, and hence, by Rule 7, $\langle b : \rightarrow q, \sigma \rangle \xrightarrow{(a)} \langle q', \sigma' \rangle$.

2. To prove the first condition, suppose $\langle p ; \bar{p}, \sigma \rangle \downarrow$. Since in any derivation with $\langle p ; \bar{p}, \sigma \rangle \downarrow$ as conclusion Rule 8 is the final rule applied, it follows that $\langle p, \sigma \rangle \downarrow$ and $\langle \bar{p}, \sigma \rangle \downarrow$. Since $\langle p, \sigma \rangle \cong_{st} \langle q, \sigma \rangle$ and $\langle \bar{p}, \sigma \rangle \cong_{st} \langle \bar{q}, \sigma \rangle$, also $\langle q, \sigma \rangle \downarrow$ and $\langle \bar{q}, \sigma \rangle \downarrow$, and hence, by Rule 8, $\langle q ; \bar{q}, \sigma \rangle \downarrow$.

To prove the second condition, suppose $\langle p ; \bar{p}, \sigma \rangle \xrightarrow{a} \langle r, \sigma' \rangle$. Note that the final rule of a derivation with this transition as conclusion is either Rule 9 or Rule 10; we treat these cases separately.

If the final rule applied is Rule 9, then it holds that $\langle p, \sigma \rangle \downarrow$, $\langle \bar{p}, \sigma \rangle \xrightarrow{a} \langle \bar{p}', \sigma' \rangle$ and $r = \bar{p}'$. Since $\langle p, \sigma \rangle \cong_{st} \langle q, \sigma \rangle$, we have $\langle q, \sigma \rangle \downarrow$. Moreover, since $\langle \bar{p}, \sigma \rangle \cong_{st} \langle \bar{q}, \sigma \rangle$, there exists $\bar{q}'$ such that $\langle \bar{q}, \sigma \rangle \xrightarrow{(a)} \langle \bar{q}', \sigma' \rangle$ and $\langle \bar{p}', \sigma' \rangle \sim_{isst} \langle \bar{q}', \sigma' \rangle$. So, by Rule 9, $\langle q ; \bar{q}, \sigma \rangle \xrightarrow{(a)} \langle q', \sigma' \rangle$.

If the final rule applied is Rule 10, then it holds that $\langle p, \sigma \rangle \xrightarrow{a} \langle p', \sigma' \rangle$ and $r = p' ; \bar{p}$. Since $\langle p, \sigma \rangle \cong_{st} \langle q, \sigma \rangle$, there exists q' such that $\langle q, \sigma \rangle \xrightarrow{(a)} \langle q', \sigma' \rangle$ and $\langle p', \sigma' \rangle \sim_{isst} \langle q', \sigma' \rangle$, so, according to Rule 10, $\langle q ; \bar{q}, \sigma \rangle \xrightarrow{(a)} \langle q' ; \bar{q}, \sigma' \rangle$. It remains to establish that $\langle p' ; \bar{p}, \sigma' \rangle \sim_{isst} \langle q' ; \bar{q}, \sigma' \rangle$, and we do this by showing that the binary relation R on configurations defined by

$$R = \sim_{isst} \cup \{(\langle r ; \bar{p}, \sigma \rangle, \langle s ; \bar{q}, \sigma \rangle) \mid \langle r, \sigma \rangle \sim_{isst} \langle s, \sigma \rangle\}$$

is an interaction sensitive stuttering bisimulation. Since $\sim_{isst}$ is an interaction sensitive stuttering bisimulation, R is symmetric and we only need to verify the conditions for pairs $(\langle r ; \bar{p}, \sigma \rangle, \langle s ; \bar{q}, \sigma \rangle)$ with $\langle r, \sigma \rangle \sim_{isst} \langle s, \sigma \rangle$.

For the termination condition, suppose $\langle r ; \bar{p}, \sigma \rangle \downarrow$. Since Rule 8 is the final rule applied in any derivation of $\langle r ; \bar{p}, \sigma \rangle \downarrow$, it holds that $\langle r, \sigma \rangle \downarrow$ and $\langle \bar{p}, \sigma \rangle \downarrow$. Since $\langle r, \sigma \rangle \sim_{isst} \langle s, \sigma \rangle$, there exist $s_0, \dots, s_n$ such that $s_0 = s$ and

$$\langle s_0, \sigma \rangle \hookrightarrow \dots \hookrightarrow \langle s_n, \sigma \rangle, \langle s_n, \sigma \rangle \downarrow, \text{ and } \langle r, \sigma \rangle \sim_{isst} \langle s_i, \sigma \rangle \text{ for all } i \leq n.$$

So by Rule 10, $\langle s_0 ; \bar{q}, \sigma \rangle \hookrightarrow \dots \hookrightarrow \langle s_n ; \bar{q}, \sigma \rangle$. Furthermore, since $\langle \bar{p}, \sigma \rangle \cong_{st} \langle \bar{q}, \sigma \rangle$ implies that also $\langle \bar{q}, \sigma \rangle \downarrow$, it follows by Rule 8 that $\langle s_n ; \bar{q}, \sigma \rangle \downarrow$, and finally, according to the definition of R , $\langle r ; \bar{p}, \sigma \rangle R \langle s_i ; \bar{q}, \sigma \rangle$ for all $i \leq n$.

For the transfer condition, suppose $\langle r; \bar{p}, \sigma \rangle \xrightarrow{a} \langle t, \sigma' \rangle$. As before, the final rule of a derivation with this transition as conclusion is either Rule 9 or Rule 10 and we treat these cases separately.

If the final rule applied is Rule 9, then it holds that $\langle r, \sigma \rangle \downarrow$, $\langle \bar{p}, \sigma \rangle \xrightarrow{a} \langle \bar{p}', \sigma' \rangle$ and $t = \bar{p}'$. Since $\langle r, \sigma \rangle \sim_{isst} \langle s, \sigma \rangle$, there exist $s_0, \dots, s_n$ such that $s_0 = s$,

$$\langle s_0, \sigma \rangle \hookrightarrow \dots \hookrightarrow \langle s_n, \sigma \rangle, \langle s_n, \sigma \rangle \downarrow \text{ and } \langle r, \sigma \rangle \sim_{isst} \langle s_i, \sigma \rangle$$

for all $0 \leq i \leq n$. So, by Rule 10,

$$\langle s_0; \bar{q}, \sigma \rangle \hookrightarrow \dots \hookrightarrow \langle s_n; \bar{q}, \sigma \rangle.$$

Furthermore, since $\langle \bar{p}, \sigma \rangle \cong_{st} \langle \bar{q}, \sigma \rangle$, there exists $\bar{q}'$ such that $\langle \bar{q}, \sigma \rangle \xrightarrow{(a)} \langle \bar{q}', \sigma' \rangle$ and $\langle \bar{p}', \sigma' \rangle \sim_{isst} \langle \bar{q}', \sigma' \rangle$. Now, by Rule 9, $\langle s_n; \bar{q}, \sigma \rangle \xrightarrow{(a)} \langle \bar{q}', \sigma' \rangle$. Finally, according to the definition of R , $\langle r; \bar{p}, \sigma \rangle R \langle s_i; \bar{q}, \sigma \rangle$ for all $i \leq n$, and $\langle t, \sigma \rangle R \langle \bar{q}', \sigma' \rangle$.

If the final rule applied is Rule 10, then it holds that $\langle r, \sigma \rangle \xrightarrow{a} \langle r', \sigma' \rangle$ and $t = r'; \bar{p}$. Since $\langle r, \sigma \rangle \sim_{isst} \langle s, \sigma \rangle$, there exist $s_0, \dots, s_n$ such that $s_0 = s$,

$$\langle s_0, \sigma \rangle \hookrightarrow \dots \hookrightarrow \langle s_{n-1}, \sigma \rangle \xrightarrow{(a)} \langle s_n, \sigma' \rangle, \langle r, \sigma \rangle \sim_{isst} \langle s_i, \sigma \rangle$$

for all $i \leq n-1$, and $\langle r', \sigma' \rangle \sim_{isst} \langle s_n, \sigma' \rangle$. So, by Rule 10,

$$\langle s_0; \bar{q}, \sigma \rangle \hookrightarrow \dots \hookrightarrow \langle s_{n-1}; \bar{q}, \sigma \rangle \xrightarrow{(a)} \langle s_n; \bar{q}, \sigma' \rangle,$$

and, according to the definition of R , $\langle r; \bar{p}, \sigma \rangle R \langle s_i; \bar{q}, \sigma \rangle$ for all $i \leq n-1$, and $\langle t, \sigma \rangle R \langle s_n; \bar{q}, \sigma' \rangle$.

To prove the divergence condition, suppose $t_0 = r; \bar{p}$,

$$\langle t_0, \sigma \rangle \hookrightarrow \langle t_1, \sigma \rangle \hookrightarrow \langle t_2, \sigma \rangle \hookrightarrow \dots,$$

and $\langle t_i, \sigma \rangle \sim_{isst} \langle s, \sigma \rangle$ for all $i \geq 0$. From the definition of R it follows that there exist $r_0, r_1, r_2, \dots$ such that $r_0 = r$, $t_i = r_i; \bar{p}$ and $\langle r_i, \sigma \rangle \sim_{isst} \langle s, \sigma \rangle$ for all $i \geq 0$. Rules 9 and 10 are the only final rules of a derivation with $\langle t_i, \sigma \rangle \hookrightarrow \langle t_{i+1}, \sigma \rangle$ as conclusion so we conclude that either $\langle r_i, \sigma \rangle \hookrightarrow \langle r_{i+1}, \sigma \rangle$ or $\langle r_i, \sigma \rangle \downarrow$ and $\langle \bar{p}, \sigma \rangle \rightarrow \langle r_{i+1}; \bar{p}, \sigma \rangle$.

Suppose first that $\langle r_i, \sigma \rangle \hookrightarrow \langle r_{i+1}, \sigma \rangle$ for all $i \geq 0$ or i.e. that $\langle r_0, \sigma \rangle \hookrightarrow \langle r_1, \sigma \rangle \hookrightarrow \langle r_2, \sigma \rangle \hookrightarrow \dots$. Since $\langle r, \sigma \rangle \sim_{isst} \langle s, \sigma \rangle$, there exist s' and $j > 0$ such that $\langle s, \sigma \rangle \hookrightarrow \langle s', \sigma \rangle$ and $\langle r_j, \sigma \rangle \sim_{isst} \langle s', \sigma \rangle$. So by Rule 10, $\langle s; \bar{q}, \sigma \rangle \hookrightarrow \langle s'; \bar{q}, \sigma \rangle$. According to the definition of R , $\langle r_j; \bar{p}, \sigma \rangle R \langle s'; \bar{q}, \sigma \rangle$.

Suppose now that $\langle r_0, \sigma \rangle \hookrightarrow \dots \hookrightarrow \langle r_n, \sigma \rangle$, $\langle r_n, \sigma \rangle \downarrow$ and $\langle \bar{p}, \sigma \rangle \rightarrow \langle r_{n+1}; \bar{p}, \sigma \rangle$. Since $\langle r_n, \sigma \rangle \sim_{isst} \langle s, \sigma \rangle$, there exist $s_0, \dots, s_m$ such that $s_0 = s$,

$$\langle s_0, \sigma \rangle \hookrightarrow \dots \hookrightarrow \langle s_m, \sigma \rangle \langle s_m, \sigma \rangle \downarrow$$

and $\langle r_n, \sigma \rangle \sim_{isst} \langle s_i, \sigma \rangle$ for all $i \leq m$. So by Rule 10,

$$\langle s_0; \bar{q}, \sigma \rangle \hookrightarrow \dots \hookrightarrow \langle s_m; \bar{q}, \sigma \rangle.$$

Furthermore, since $\langle \bar{p}, \sigma \rangle \cong_{st} \langle \bar{q}, \sigma \rangle$, there exists $\bar{q}'$ such that $\langle \bar{q}, \sigma \rangle \hookrightarrow \langle \bar{q}', \sigma \rangle$ and $\langle r_{n+1}; \bar{p}, \sigma \rangle \sim_{isst} \langle \bar{q}', \sigma \rangle$, and hence, by Rule 9, $\langle s_m; \bar{q}, \sigma \rangle \hookrightarrow \langle \bar{q}', \sigma \rangle$. We now distinguish three cases and show for each that the divergence condition is satisfied. If $m > 0$ and $n > 0$ then, $\langle r_n; \bar{p}, \sigma \rangle R \langle s_1; \bar{q}, \sigma \rangle$. If $m > 0$ but $n = 0$, using that $\langle r_0, \sigma \rangle \sim_{isst} \langle s_0, \sigma \rangle$, $\langle r_0, \sigma \rangle \sim_{isst} \langle s_1, \sigma \rangle$, $\langle r_1, \sigma \rangle \sim_{isst} \langle s_0, \sigma \rangle$, and that $\sim_{isst}$ is transitive, we conclude that $\langle r_1, \sigma \rangle \sim_{isst} \langle s_1, \sigma \rangle$, and hence, that $\langle r_1; \bar{p}, \sigma \rangle R \langle s_1; \bar{q}, \sigma \rangle$. The last case is when $m = 0$. In this case, $\langle r_{n+1}; \bar{p}, \sigma \rangle R \langle \bar{q}', \sigma \rangle$.

Since R satisfies all the conditions from Definition , we conclude that R is an interaction sensitive stuttering bisimulation.

3. To prove the first condition, suppose $\langle p \parallel \bar{p}, \sigma \rangle \downarrow$. As Rule 11 is the final rule of any derivation with this as conclusion, it holds that $\langle p, \sigma \rangle \downarrow$ or $\langle \bar{p}, \sigma \rangle \downarrow$. We only consider the first case; when $\langle \bar{p}, \sigma \rangle \downarrow$ the proof is similar. Since $\langle p, \sigma \rangle \cong_{st} \langle q, \sigma \rangle$, it follows that $\langle q, \sigma \rangle \downarrow$, and hence, by Rule 11, that $\langle q \parallel \bar{q}, \sigma \rangle \downarrow$. To prove the second condition, suppose $\langle p \parallel \bar{p}, \sigma \rangle \xrightarrow{a} \langle r, \sigma' \rangle$. Since Rule 12 must be the final Rule of any derivation of this transition, it holds that $\langle p, \sigma \rangle \xrightarrow{a} \langle p', \sigma' \rangle$ and $r = p'$ or that $\langle \bar{p}, \sigma \rangle \xrightarrow{a} \langle \bar{p}', \sigma' \rangle$ and $r = \bar{p}'$. Suppose $\langle p, \sigma \rangle \xrightarrow{a} \langle p', \sigma' \rangle$ (the other case is symmetric). Since $\langle p, \sigma \rangle \cong_{st} \langle q, \sigma \rangle$, there exists q' such that $\langle q, \sigma \rangle \xrightarrow{(a)} \langle q', \sigma' \rangle$ and $\langle p', \sigma' \rangle \cong_{st} \langle q', \sigma' \rangle$. Now, by Rule 12, $\langle q \parallel \bar{q}, \sigma \rangle \xrightarrow{(a)} \langle q', \sigma' \rangle$.
4. By Rule 13, the first condition of rooted interaction sensitive stuttering bisimilarity holds trivially. For the second condition, suppose $\langle p^*, \sigma \rangle \xrightarrow{a} \langle r, \sigma' \rangle$. Since in any derivation with this transition Rule 14 is the final rule applied, it follows that $\langle p, \sigma \rangle \xrightarrow{a} \langle p', \sigma' \rangle$ and $r = p'; p^*$. Since $\langle p, \sigma \rangle \cong_{st} \langle q, \sigma \rangle$, there exists q' such that $\langle q, \sigma \rangle \xrightarrow{(a)} \langle q', \sigma' \rangle$ and $\langle p', \sigma' \rangle \sim_{isst} \langle q', \sigma' \rangle$. By Rule 14, $\langle q^*, \sigma \rangle \xrightarrow{(a)} \langle q'; q^*, \sigma' \rangle$. It remains to establish that $\langle r, \sigma' \rangle \sim_{isst} \langle q'; q^*, \sigma' \rangle$. We do this by showing that a symmetric relation

$$R = \{(\langle r; p^*, \sigma \rangle, \langle s; q^*, \sigma \rangle) \mid \langle r, \sigma \rangle \sim_{isst} \langle s, \sigma \rangle\}$$

is a interaction sensitive stuttering bisimulation.

To prove the termination condition, suppose $\langle r; p^*, \sigma \rangle \downarrow$. Since Rule 8 is the final rule applied in any derivation of $\langle r; p^*, \sigma \rangle \downarrow$, and since $\langle p^*, \sigma \rangle \downarrow$ by Rule 13, it holds that $\langle r, \sigma \rangle \downarrow$. Since $\langle r, \sigma \rangle \sim_{isst} \langle s, \sigma \rangle$, there exist $s_0, \dots, s_n$ such that $s_0 = s$,

$$\langle s_0, \sigma \rangle \hookrightarrow \dots \hookrightarrow \langle s_n, \sigma \rangle, \quad \langle s_n, \sigma \rangle \downarrow,$$

and $\langle r, \sigma \rangle \sim_{isst} \langle s_i, \sigma \rangle$ for all $i \leq n$. By applying Rule 10 now, we obtain

$$\langle s_0; q^*, \sigma \rangle \hookrightarrow \dots \hookrightarrow \langle s_n; q^*, \sigma \rangle,$$

and hence, by Rules 8 and 13 again, $\langle s_n; q^*, \sigma \rangle \downarrow$. According to the definition of R , $\langle r; p^*, \sigma \rangle R \langle s_i; q^*, \sigma \rangle$ for all $i \leq n$.

To prove the transfer condition, suppose $\langle r; p^*, \sigma \rangle \xrightarrow{a} \langle t, \sigma' \rangle$. Note that the final rule of a derivation with this transition as conclusion is either Rule 9 or Rule 10; we treat these cases separately.

If the final rule applied is Rule 9, then it holds that $\langle r, \sigma \rangle \downarrow$ and $\langle p^*, \sigma \rangle \xrightarrow{a} \langle t, \sigma' \rangle$. Since $\langle r, \sigma \rangle \sim_{isst} \langle s, \sigma \rangle$, there exist $s_0, \dots, s_n$ such that $s_0 = s$,

$$\langle s_0, \sigma \rangle \hookrightarrow \dots \hookrightarrow \langle s_n, \sigma \rangle, \langle s_n, \sigma \rangle \downarrow,$$

and $\langle r, \sigma \rangle \sim_{isst} \langle s_i, \sigma \rangle$ for all $i \leq n$. Applying Rule 10,

$$\langle s_0; q^*, \sigma \rangle \hookrightarrow \dots \hookrightarrow \langle s_n; q^*, \sigma \rangle.$$

Furthermore, since Rule 14 is the final rule of a derivation with $\langle p^*, \sigma \rangle \xrightarrow{a} \langle t, \sigma' \rangle$ as conclusion, we have $\langle p, \sigma \rangle \xrightarrow{a} \langle p', \sigma' \rangle$ and $t = p'; p^*$. Since $\langle p, \sigma \rangle \cong_{st} \langle q, \sigma \rangle$, there exists q' such that $\langle q, \sigma \rangle \xrightarrow{(a)} \langle q', \sigma' \rangle$ and $\langle p', \sigma' \rangle \sim_{isst} \langle q', \sigma' \rangle$. Applying Rule 14, we obtain $\langle q^*, \sigma \rangle \xrightarrow{(a)} \langle q'; q^*, \sigma' \rangle$. Since $\langle s_n, \sigma \rangle \downarrow$, by Rule 9, we have $\langle s_n; q^*, \sigma \rangle \xrightarrow{(a)} \langle q'; q^*, \sigma' \rangle$. Finally, according to the definition of R , $\langle r; p^*, \sigma \rangle R \langle s_i; q^*, \sigma \rangle$ for all $i \leq n$, and $\langle t, \sigma' \rangle R \langle q'; q^*, \sigma' \rangle$.

If the final rule applied is Rule 10, then it holds that $\langle r, \sigma \rangle \xrightarrow{a} \langle r', \sigma' \rangle$ and $t = r'; p^*$. Since $\langle r, \sigma \rangle \sim_{isst} \langle s, \sigma \rangle$, there exist $s_0, \dots, s_n$ such that $s_0 = s$,

$$\langle s_0, \sigma \rangle \hookrightarrow \dots \hookrightarrow \langle s_{n-1}, \sigma \rangle \xrightarrow{(a)} \langle s_n, \sigma' \rangle,$$

$\langle r, \sigma \rangle \sim_{isst} \langle s_i, \sigma \rangle$ for all $i \leq n-1$, and $\langle r', \sigma' \rangle \sim_{isst} \langle s_n, \sigma' \rangle$. Now, applying Rule 6, we obtain

$$\langle s_0; q^*, \sigma \rangle \hookrightarrow \dots \hookrightarrow \langle s_{n-1}; q^*, \sigma \rangle \xrightarrow{(a)} \langle s_n; q^*, \sigma' \rangle,$$

and according to the definition of R , $\langle r; p^*, \sigma \rangle R \langle s_i; q^*, \sigma \rangle$ for all $i \leq n-1$, and $\langle t, \sigma' \rangle R \langle s_n; q^*, \sigma' \rangle$.

Since the divergence condition can be proved similarly as in the case of the sequential operator, we conclude that R is an interaction sensitive stuttering bisimulation.

5. To prove the first condition, suppose $\langle [s \mid p], \sigma \rangle \downarrow$. Since in any derivation with this as conclusion Rule 18 is the final rule applied, we have $\langle p, \gamma(s, \sigma) \rangle \downarrow$. Since $\langle p, \gamma(s, \sigma) \rangle \downarrow \cong_{st} \langle q, \gamma(s, \sigma) \rangle \downarrow$, it follows that $\langle q, \gamma(s, \sigma) \rangle \downarrow$, and hence, by Rule 18, $\langle [s \mid q], \sigma \rangle \downarrow$.

To prove the second condition, suppose $\langle [s \mid p], \sigma \rangle \xrightarrow{a} \langle r, \sigma'' \rangle$. Since Rule 19 is the final rule of any derivation with this transition as conclusion, it holds that $\langle p, \gamma(s, \sigma) \rangle \xrightarrow{a} \langle p', \sigma' \rangle$, $\sigma'' = \gamma(\sigma' \upharpoonright \text{dom}(\sigma) \setminus \text{dom}(s), \sigma)$, $r = [s' \mid p']$ and $s' = \sigma' \upharpoonright \text{dom}(s)$. Since $\langle p, \gamma(s, \sigma) \rangle \cong_{st} \langle q, \gamma(s, \sigma) \rangle$, there exists q' such that $\langle q, \gamma(s, \sigma) \rangle \xrightarrow{(a)} \langle q', \sigma' \rangle$ and $\langle p', \sigma' \rangle \sim_{isst} \langle q', \sigma' \rangle$. Hence, by Rule 19, we obtain $\langle [s \mid q], \sigma \rangle \xrightarrow{(a)} \langle [s' \mid q'], \sigma'' \rangle$. It is not hard to show that

$$\gamma(\sigma' \upharpoonright \text{dom}(st), \gamma(\sigma' \upharpoonright \text{dom}(\sigma) \setminus \text{dom}(st), \sigma)) = \sigma'$$

so, to prove that $\langle [s' \mid p'], \sigma'' \rangle \sim_{isst} \langle [s' \mid q'], \sigma'' \rangle$, it is enough to show that a symmetric binary relation

$$R = \{(\langle [st \mid r], \sigma \rangle, \langle [st \mid s], \sigma \rangle) \mid st \in \Sigma, \langle r, \gamma(st, \sigma) \rangle \sim_{isst} \langle s, \gamma(st, \sigma) \rangle\}$$

is an interaction sensitive stuttering bisimulation.

To prove the termination condition, suppose $\langle \llbracket st \mid r \rrbracket, \sigma \rangle \downarrow$. Since Rule 18 is the final rule of any derivation with this as conclusion, it holds that $\langle r, \gamma(st, \sigma) \rangle \downarrow$. Since $\langle r, \gamma(st, \sigma) \rangle \sim_{isst} \langle s, \gamma(st, \sigma) \rangle$, there exist $s_0, \dots, s_n$ such that $s_0 = s$,

$$\langle s_0, \gamma(st, \sigma) \rangle \hookrightarrow \dots \hookrightarrow \langle s_n, \gamma(st, \sigma) \rangle, \langle s_n, \gamma(st, \sigma) \rangle \downarrow,$$

and $\langle r, \gamma(st, \sigma) \rangle \sim_{isst} \langle s_i, \gamma(st, \sigma) \rangle$ for all $i \leq n$. Hence, by Rule 18, $\langle \llbracket st \mid s_n \rrbracket, \sigma \rangle \downarrow$. Using that $\gamma(st, \sigma) \upharpoonright \text{dom}(st) = st$ and $\gamma(\gamma(st, \sigma) \upharpoonright \text{dom}(\sigma) \setminus \text{dom}(s), \sigma) = \sigma$, by Rule 19, we obtain

$$\langle \llbracket st \mid s_0 \rrbracket, \sigma \rangle \hookrightarrow \langle \llbracket st \mid s_1 \rrbracket, \sigma \rangle \hookrightarrow \dots \hookrightarrow \langle \llbracket st \mid s_n \rrbracket, \sigma \rangle.$$

Finally, according to the definition of R , $\langle \llbracket st \mid r \rrbracket, \sigma \rangle R \langle \llbracket st \mid s_i \rrbracket, \sigma \rangle$ for all $i \leq n$.

To prove the transfer condition, suppose $\langle \llbracket st \mid r \rrbracket, \sigma \rangle \xrightarrow{a} \langle t, \sigma'' \rangle$. Since Rule 19 is the final rule with this transition as conclusion, we have $\langle r, \gamma(st, \sigma) \rangle \xrightarrow{a} \langle r', \sigma' \rangle$, $\sigma'' = \gamma(\sigma, \sigma' \upharpoonright \text{dom}(\sigma) \setminus \text{dom}(st))$, $t = \llbracket st' \mid r' \rrbracket$, and $st' = \sigma' \upharpoonright \text{dom}(st)$. Since $\langle r, \gamma(st, \sigma) \rangle \sim_{isst} \langle s, \gamma(st, \sigma) \rangle$, there exist $s_0, \dots, s_n$ such that $s_0 = s$,

$$\langle s_0, \gamma(st, \sigma) \rangle \hookrightarrow \dots \hookrightarrow \langle s_{n-1}, \gamma(st, \sigma) \rangle \xrightarrow{(a)} \langle s_n, \sigma' \rangle,$$

$\langle r, \gamma(st, \sigma) \rangle \sim_{isst} \langle s_i, \gamma(st, \sigma) \rangle$ for all $i \leq n$, and $\langle r', \sigma' \rangle \sim_{isst} \langle s_n, \sigma' \rangle$. Now, applying Rule 19,

$$\langle \llbracket st \mid s_0 \rrbracket, \sigma \rangle \hookrightarrow \dots \hookrightarrow \langle \llbracket st \mid s_{n-1} \rrbracket, \sigma \rangle \xrightarrow{(a)} \langle \llbracket st' \mid s_n \rrbracket, \sigma'' \rangle.$$

According to the definition of R it follows that $\langle \llbracket st \mid r \rrbracket, \sigma \rangle R \langle \llbracket st \mid s_i \rrbracket, \sigma \rangle$ for all $i \leq n$, and $\langle \llbracket st' \mid r' \rrbracket, \sigma \rangle R \langle \llbracket st' \mid s_n \rrbracket, \sigma \rangle$.

To prove the divergence condition, suppose $t_0 = \llbracket st \mid r \rrbracket$, $\langle t_0, \sigma \rangle \hookrightarrow \langle t_1, \sigma \rangle \hookrightarrow \langle t_2, \sigma \rangle \hookrightarrow \dots$ and $\langle \llbracket st \mid s \rrbracket, \sigma \rangle R \langle t_i, \sigma \rangle$ for all $i \geq 0$. Since Rule 19 is the final rule with $\langle t_i, \sigma \rangle \hookrightarrow \langle t_{i+1}, \sigma \rangle$ as conclusion, we have $r_0 = r$,

$$\langle r_0, \gamma(st, \sigma) \rangle \hookrightarrow \langle r_1, \gamma(st, \sigma) \rangle \hookrightarrow \langle r_2, \gamma(st, \sigma) \rangle \hookrightarrow \dots$$

According to the definition of R , we also have $\langle r_i, \gamma(st, \sigma) \rangle \sim_{isst} \langle s, \gamma(st, \sigma) \rangle$ for all $i \geq 0$. From the divergence condition of $\sim_{isst}$ it follows that there exist s' and $j > 0$ such that $\langle s, \gamma(st, \sigma) \rangle \hookrightarrow \langle s', \gamma(st, \sigma) \rangle$ and $\langle r_j, \gamma(st, \sigma) \rangle \sim_{isst} \langle s', \gamma(st, \sigma) \rangle$, and hence, by Rule 19, $\llbracket st \mid s \rrbracket \hookrightarrow \llbracket st \mid s' \rrbracket$. According to the definition of R , $\langle \llbracket st \mid r_j \rrbracket, \sigma \rangle R \langle \llbracket st \mid s' \rrbracket, \sigma \rangle$.

We conclude that R is an interaction sensitive stuttering bisimulation.

6. To prove the first condition, suppose $\langle p \parallel \bar{p}, \sigma \rangle \downarrow$. Since Rule 15 is the final rule with this as conclusion, we have $\langle p, \sigma \rangle \downarrow$ and $\langle \bar{p}, \sigma \rangle \downarrow$. Since $\langle p, \sigma \rangle \cong_{st} \langle q, \sigma \rangle$ and $\langle \bar{p}, \sigma \rangle \cong_{st} \langle \bar{q}, \sigma \rangle$, we obtain $\langle q, \sigma \rangle \downarrow$ and $\langle \bar{q}, \sigma \rangle \downarrow$, and hence, by Rule 15 again, $\langle q \parallel \bar{q}, \sigma \rangle \downarrow$.

To prove the second condition, suppose $\langle p \parallel \bar{p}, \sigma \rangle \xrightarrow{a} \langle r, \sigma' \rangle$. The final rule of a derivation with this transition as conclusion is either Rule 16 or Rule 17; we treat these cases separately.

If the final rule applied is Rule 16, then $\langle p, \sigma \rangle \xrightarrow{a} \langle p', \sigma' \rangle$ and $t = p' \parallel \bar{p}$; (or symmetrically $\langle \bar{p}, \sigma \rangle \xrightarrow{a} \langle \bar{p}', \sigma' \rangle$ and $t = p \parallel \bar{p}'$). Since $\langle p, \sigma \rangle \cong_{st} \langle q, \sigma \rangle$, there exists q' such that $\langle q, \sigma \rangle \xrightarrow{(a)} \langle q', \sigma' \rangle$ and $\langle p', \sigma' \rangle \sim_{isst} \langle q', \sigma' \rangle$. Hence, by Rule 16, $\langle q \parallel \bar{q}, \sigma \rangle \xrightarrow{(a)} \langle q' \parallel \bar{q}, \sigma' \rangle$.

If the final rule applied is Rule 17, we have that

$$\langle p, \sigma \rangle \xrightarrow{sa(m,c)} \langle p', \sigma \rangle \text{ and } \langle \bar{p}, \sigma \rangle \xrightarrow{ra(m,c)} \langle \bar{p}', \sigma' \rangle,$$

with $a = ca(m, c)$ and $t = p' \parallel \bar{p}'$. Since $\langle p, \sigma \rangle \cong_{st} \langle q, \sigma \rangle$ and $\langle \bar{p}, \sigma \rangle \cong_{st} \langle \bar{q}, \sigma \rangle$, there exist q' and $\bar{q}'$ such that

$$\langle q, \sigma \rangle \xrightarrow{sa(m,c)} \langle q', \sigma \rangle \text{ and } \langle \bar{q}, \sigma \rangle \xrightarrow{ra(m,c)} \langle \bar{q}', \sigma' \rangle,$$

and $\langle p', \sigma \rangle \sim_{isst} \langle q', \sigma \rangle$ and $\langle \bar{p}', \sigma' \rangle \sim_{isst} \langle \bar{q}', \sigma' \rangle$. Now, by Rule 17,

$$\langle q \parallel \bar{q}, \sigma \rangle \xrightarrow{ca(m,c)} \langle q' \parallel \bar{q}', \sigma' \rangle.$$

It remains to show that $\langle p' \parallel \bar{p}', \sigma' \rangle \sim_{isst} \langle q' \parallel \bar{q}', \sigma' \rangle$, and for this it is enough to show that the symmetric relation

$$R = \{(\langle r \parallel \bar{r}, \sigma \rangle, \langle s \parallel \bar{s}, \sigma \rangle) \mid \langle r, \sigma \rangle \sim_{isst} \langle s, \sigma \rangle, \langle \bar{r}, \sigma \rangle \sim_{isst} \langle \bar{s}, \sigma \rangle\}$$

is an interaction sensitive stuttering bisimulation.

To prove the termination condition, suppose $\langle r \parallel \bar{r}, \sigma \rangle \downarrow$. Since Rule 15 is the final rule with this as conclusion, we have $\langle r, \sigma \rangle \downarrow$ and $\langle \bar{r}, \sigma \rangle \downarrow$. Since $\langle r, \sigma \rangle \sim_{isst} \langle s, \sigma \rangle$ there exist $s_0, \dots, s_n$ such that $s_0 = s$,

$$\langle s_0, \sigma \rangle \hookrightarrow \dots \hookrightarrow \langle s_n, \sigma \rangle, \langle s_n, \sigma \rangle \downarrow,$$

and $\langle r, \sigma \rangle \sim_{isst} \langle s_i, \sigma \rangle$ for all $i \leq n$. Since $\langle \bar{r}, \sigma \rangle \sim_{isst} \langle \bar{s}, \sigma \rangle$, there exist $\bar{s}_0, \dots, \bar{s}_m$ such that $\bar{s}_0 = \bar{s}$,

$$\langle \bar{s}_0, \sigma \rangle \hookrightarrow \dots \hookrightarrow \langle \bar{s}_m, \sigma \rangle, \langle \bar{s}_m, \sigma \rangle \downarrow,$$

and $\langle \bar{r}, \sigma \rangle \sim_{isst} \langle \bar{s}_i, \sigma \rangle$ for all $i \leq m$. By Rule 16,

$$\langle s_0 \parallel \bar{s}_0, \sigma \rangle \hookrightarrow \dots \hookrightarrow \langle s_n \parallel \bar{s}_0, \sigma \rangle \hookrightarrow \dots \hookrightarrow \langle s_n \parallel \bar{s}_m, \sigma \rangle,$$

and by Rule 15 $\langle s_n \parallel \bar{s}_m, \sigma \rangle \downarrow$. According to the definition of R , it holds that $\langle r \parallel \bar{r}, \sigma \rangle R \langle s_i \parallel \bar{s}_j, \sigma \rangle$ for all $i \leq n, j \leq m$.

To prove the transfer condition, suppose $\langle r \parallel \bar{r}, \sigma \rangle \xrightarrow{a} \langle t, \sigma' \rangle$. The final rule of a derivation with this transition as conclusion is either Rule 16 or Rule 17; we treat these cases separately.

If the final rule applied is Rule 16, then $\langle r, \sigma \rangle \xrightarrow{a} \langle r', \sigma \rangle$ and $t = r' \parallel \bar{r}$ (or, symmetrically $\langle \bar{r}, \sigma \rangle \xrightarrow{a} \langle \bar{r}', \sigma \rangle$ and $t = r \parallel \bar{r}'$). Since $\langle r, \sigma \rangle \sim_{isst} \langle s, \sigma \rangle$, there exist $s_0, \dots, s_n$ such that $s_0 = s$,

$$\langle s_0, \sigma \rangle \hookrightarrow \dots \hookrightarrow \langle s_{n-1}, \sigma \rangle \xrightarrow{(a)} \langle s_n, \sigma' \rangle,$$

$\langle r, \sigma \rangle \sim_{isst} \langle s_i, \sigma \rangle$ for all $i \leq n-1$, and $\langle r', \sigma' \rangle \sim_{isst} \langle s_n, \sigma' \rangle$. Applying Rule 16,

$$\langle s_0 \parallel \bar{s}, \sigma \rangle \hookrightarrow \dots \hookrightarrow \langle s_{n-1} \parallel \bar{s}, \sigma \rangle \xrightarrow{(a)} \langle s_n \parallel \bar{s}, \sigma' \rangle,$$

and according to the definition of R , $\langle r \parallel \bar{r}, \sigma \rangle R \langle s_i \parallel \bar{s}, \sigma \rangle$ for all $i \leq n-1$, and $\langle t, \sigma \rangle R \langle s_n \parallel \bar{s}, \sigma \rangle$.

If the final rule applied is Rule 17, $\langle r, \sigma \rangle \xrightarrow{sa(m,c)} \langle r', \sigma \rangle$, $\langle \bar{r}, \sigma \rangle \xrightarrow{ra(m,c)} \langle \bar{r}', \sigma' \rangle$, $t = r' \parallel \bar{r}'$ and $a = ca(m, c)$ (or, symmetrically, $\bar{r}$ does send and r does receive). Since $\langle r, \sigma \rangle \sim_{st} \langle s, \sigma \rangle$, there exist $s_0, \dots, s_n, s'$ such that $s_0 = s$,

$$\langle s_0, \sigma \rangle \hookrightarrow \dots \hookrightarrow \langle s_n, \sigma \rangle \xrightarrow{sa(m,c)} \langle s', \sigma' \rangle,$$

$\langle r, \sigma \rangle \sim_{isst} \langle s_i, \sigma \rangle$ for all $i \leq n$, and $\langle r', \sigma \rangle \sim_{isst} \langle s', \sigma' \rangle$. Since $\langle \bar{r}, \sigma \rangle \sim_{isst} \langle \bar{s}, \sigma \rangle$, there exist $\bar{s}_0, \dots, \bar{s}_m, \bar{s}'$ such that $\bar{s}_0 = \bar{s}$,

$$\langle \bar{s}_0, \sigma \rangle \hookrightarrow \dots \hookrightarrow \langle \bar{s}_m, \sigma \rangle \xrightarrow{ra(m,c)} \langle \bar{s}', \sigma' \rangle,$$

$\langle r, \sigma \rangle \sim_{isst} \langle \bar{s}_i, \sigma \rangle$ for all $i \leq m$, and $\langle r', \sigma' \rangle \sim_{isst} \langle \bar{s}', \sigma' \rangle$. By applying Rule 16, $\langle s_0 \parallel \bar{s}_0, \sigma \rangle \hookrightarrow \dots \hookrightarrow \langle s_n \parallel \bar{s}_0, \sigma \rangle \hookrightarrow \dots \hookrightarrow \langle s_n \parallel \bar{s}_m, \sigma \rangle$. By Rule 17,

$$\langle s_n \parallel \bar{s}_m, \sigma \rangle \xrightarrow{ca(m,c)} \langle s' \parallel \bar{s}', \sigma' \rangle.$$

According to the definition of R , $\langle r \parallel \bar{r}, \sigma \rangle R \langle s_i \parallel \bar{s}_j, \sigma \rangle$ for all $i \leq n, j \leq m$, and $\langle t, \sigma \rangle R \langle s' \parallel \bar{s}', \sigma' \rangle$.

To prove the divergence condition, suppose $t_0 = r \parallel \bar{r}$,

$$\langle t_0, \sigma \rangle \hookrightarrow \langle t_1, \sigma \rangle \hookrightarrow \langle t_2, \sigma \rangle \hookrightarrow \dots$$

and $\langle s \parallel \bar{s}, \sigma \rangle R \langle t_i, \sigma \rangle$ for all $i \geq 0$. According to the definition of R there exist $r_0, r_1, r_2, \dots$ and $\bar{r}_0, \bar{r}_1, \bar{r}_2, \dots$ such that $t_i = r_i \parallel \bar{r}_i$, $\langle r_i, \sigma \rangle \sim_{isst} \langle s, \sigma \rangle$ and $\langle \bar{r}_i, \sigma \rangle \sim_{isst} \langle \bar{s}, \sigma \rangle$. Since the final rule of a derivation with $\langle r_i \parallel \bar{r}_i, \sigma \rangle \hookrightarrow \langle t_{i+1}, \sigma \rangle$ as conclusion is either Rule 16 or Rule 17, we conclude that for all $i \geq 0$ either $\langle r_i, \sigma \rangle \hookrightarrow \langle r_{i+1}, \sigma \rangle$ and $\bar{r}_{i+1} = \bar{r}_i$, or $r_i \xrightarrow{sa(m,c)} r_{i+1}$ and $\bar{r}_i \xrightarrow{ra(m,c)} \bar{r}_{i+1}$ (or the two symmetric cases).

Suppose that there exist n and m such that $r_i \not\rightarrow r_{i+1}$ and $\bar{r}_j \not\rightarrow \bar{r}_{j+1}$ for all $i \geq n, j \geq m$ (note that this implies that for all $i \geq n$ and for all $j \geq m$, $r_{i+1} = r_i$ and $\bar{r}_{j+1} = \bar{r}_j$). Let $k = \max(n, m)$. Then, $\langle t_k, \sigma \rangle \not\rightarrow \langle t_{k+1}, \sigma \rangle$ which is a contradiction. We conclude that there exist $i_0, i_1, i_2, \dots$ such that $i_0 = i$,

$$\langle r_{i_0}, \sigma \rangle \rightarrow \langle r_{i_1}, \sigma \rangle \rightarrow \langle r_{i_2}, \sigma \rangle \rightarrow \dots$$

or that there exist $j_0, j_1, j_2, \dots$ such that $j_0 = j$,

$$\langle \bar{r}_{j_0}, \sigma \rangle \rightarrow \langle \bar{r}_{j_1}, \sigma \rangle \rightarrow \langle \bar{r}_{j_2}, \sigma \rangle \rightarrow \dots$$

We only consider the first case, the second one is similar.

Suppose first $\langle r_{i_0}, \sigma \rangle \hookrightarrow \langle r_{i_1}, \sigma \rangle \hookrightarrow \langle r_{i_2}, \sigma \rangle \hookrightarrow \dots$. Since $\langle r_{i_j}, \sigma \rangle \sim_{isst} \langle s, \sigma \rangle$ for all $j \geq 0$, there are s' and $k > 0$ such that $\langle s, \sigma \rangle \hookrightarrow \langle s', \sigma \rangle$ and $\langle r_{i_k}, \sigma \rangle \sim_{isst} \langle s', \sigma \rangle$. By Rule 16, $\langle s \parallel \bar{s}, \sigma \rangle \hookrightarrow \langle s' \parallel \bar{s}, \sigma \rangle$ and according to the definition of R , $\langle t_{i_k}, \sigma \rangle R \langle s' \parallel \bar{s}, \sigma \rangle$.

Suppose now $\langle r_{i_0}, \sigma \rangle \hookrightarrow \dots \hookrightarrow \langle r_{i_k}, \sigma \rangle \xrightarrow{a} \langle r_{i_{k+1}}, \sigma \rangle$ for some $k \geq 0$ and $a \in \mathcal{A}^{com}$. Since $\langle r_{i_k}, \sigma \rangle \sim_{isst} \langle s, \sigma \rangle$, there exist $s_0, \dots, s_n, s'$ such that $s_0 = s$,

$$\langle s_0, \sigma \rangle \hookrightarrow \dots \hookrightarrow \langle s_n, \sigma \rangle \xrightarrow{a} \langle s', \sigma \rangle,$$

$\langle r_{i_k}, \sigma \rangle \sim_{isst} \langle s_j, \sigma \rangle$ for all $j \leq n$, and $\langle r_{i_{k+1}}, \sigma \rangle \sim_{isst} \langle s', \sigma \rangle$. Suppose $a = sa(m, c)$ (similarly if a is a receive action). Then, $\langle \bar{r}_{i_k}, \sigma \rangle \xrightarrow{ra(m, c)} \langle \bar{r}_{i_{k+1}}, \sigma \rangle$. Since $\langle \bar{r}_{i_k}, \sigma \rangle \sim_{isst} \langle \bar{s}, \sigma \rangle$, there exist $\bar{s}_0, \dots, \bar{s}_m, \bar{s}'$ such that $\bar{s}_0 = \bar{s}$,

$$\langle \bar{s}_0, \sigma \rangle \hookrightarrow \dots \hookrightarrow \langle \bar{s}_m, \sigma \rangle \xrightarrow{ra(m, c)} \langle \bar{s}', \sigma \rangle,$$

$\langle \bar{r}_{i_n}, \sigma \rangle \sim_{isst} \langle \bar{s}_j, \sigma \rangle$ for all $j \leq m$, and $\langle \bar{r}_{i_{k+1}}, \sigma \rangle \sim_{isst} \langle \bar{s}', \sigma \rangle$. By Rules 16, 17,

$$\langle s \parallel \bar{s}, \sigma \rangle \hookrightarrow \dots \hookrightarrow \langle s_n \parallel \bar{s}_m, \sigma \rangle \xrightarrow{ca(m, c)} \langle s' \parallel \bar{s}', \sigma \rangle$$

and from the definition of R , $\langle r_{i_{k+1}}, \sigma \rangle R \langle s' \parallel \bar{s}', \sigma \rangle$.

The proof that R is an interaction sensitive stuttering bisimulation relation is now complete.

7. To prove the first condition, suppose $\langle \partial(p), \sigma \rangle \downarrow$. Since Rule 20 is the final rule with this as conclusion, it follows that $\langle p, \sigma \rangle \downarrow$. Since $\langle p, \sigma \rangle \cong_{st} \langle q, \sigma \rangle$, we have $\langle q, \sigma \rangle \downarrow$, and hence, by Rule 20, that $\langle \partial(q), \sigma \rangle \downarrow$.

To prove the second condition, suppose $\langle \partial(p), \sigma \rangle \xrightarrow{a} \langle r, \sigma' \rangle$. By Rule 21, as the final rule with this transition as conclusion, we obtain $\langle p, \sigma \rangle \xrightarrow{a} \langle p', \sigma' \rangle$, $r = \partial(p')$ and $a \in \mathcal{A} \setminus \mathcal{A}^{com}$. Since $\langle p, \sigma \rangle \cong_{st} \langle q, \sigma \rangle$, there exists q' such that $\langle q, \sigma \rangle \hookrightarrow \langle q', \sigma' \rangle$ and $\langle p', \sigma' \rangle \sim_{isst} \langle q', \sigma' \rangle$. By Rule 21 again, $\langle \partial(q), \sigma \rangle \hookrightarrow \langle \partial(q'), \sigma' \rangle$. It remains to show that $\langle \partial(p'), \sigma \rangle \sim_{isst} \langle \partial(q'), \sigma' \rangle$. We do this by showing that a symmetric relation

$$R = \{(\langle \partial(r), \sigma \rangle, \langle \partial(s), \sigma \rangle) \mid \langle r, \sigma \rangle \sim_{isst} \langle s, \sigma \rangle\}$$

is an interaction sensitive stuttering bisimulation relation.

To prove the termination condition, suppose $\langle \partial(r), \sigma \rangle \downarrow$. Rule 20 is the final rule with this as conclusion so we have $\langle r, \sigma \rangle \downarrow$. Since $\langle r, \sigma \rangle \sim_{isst} \langle s, \sigma \rangle$, there exist $s_0, \dots, s_n$ such that $s_0 = s$,

$$\langle s_0, \sigma \rangle \hookrightarrow \dots \hookrightarrow \langle s_n, \sigma \rangle, \quad \langle s_n, \sigma \rangle \downarrow,$$

and $\langle r, \sigma \rangle \sim_{isst} \langle s_i, \sigma \rangle$ for all $i \leq n$. Now, by Rules 20 and 21,

$$\langle \partial(s_0), \sigma \rangle \hookrightarrow \dots \hookrightarrow \langle \partial(s_n), \sigma \rangle$$

and $\langle \partial(s_n), \sigma \rangle \downarrow$. According to the definition of R , $\langle \partial(r), \sigma \rangle R \langle \partial(s_i), \sigma \rangle$ for all $i \leq n$.

To prove the transfer condition, suppose $\langle \partial(r), \sigma \rangle \xrightarrow{a} \langle t, \sigma' \rangle$. Since Rule 21 is the final rule with this transition as conclusion, we have $\langle r, \sigma \rangle \xrightarrow{a} \langle r', \sigma' \rangle$, $t = \partial(r')$ and $a \in \mathcal{A} \setminus \mathcal{A}^{com}$. Since $\langle r, \sigma \rangle \sim_{isst} \langle s, \sigma \rangle$ there exist $s_0, \dots, s_n$ such that $s_0 = s$,

$$\langle s_0, \sigma \rangle \hookrightarrow \dots \hookrightarrow \langle s_{n-1}, \sigma \rangle \hookrightarrow \langle s_n, \sigma \rangle,$$

$\langle r, \sigma \rangle \cong_{st} \langle s_i, \sigma \rangle$ for all $i \leq n-1$, and $\langle r', \sigma' \rangle \sim_{isst} \langle s_n, \sigma' \rangle$. Now, by applying Rule 21,

$$\langle \partial(s_0), \sigma \rangle \hookrightarrow \dots \hookrightarrow \langle \partial(s_{n-1}), \sigma \rangle \hookrightarrow \langle \partial(s_n), \sigma' \rangle.$$

According to the definition of R , $\langle \partial(r), \sigma \rangle R \langle \partial(s_i), \sigma \rangle$ for all $i \leq n-1$, and $\langle \partial(r'), \sigma' \rangle R \langle \partial(s_n), \sigma' \rangle$.

To prove the divergence condition, suppose $t_0 = \partial(r)$, $\langle t_0, \sigma \rangle \hookrightarrow \langle t_1, \sigma \rangle \hookrightarrow \langle t_2, \sigma \rangle \hookrightarrow \dots$ and $\langle t_i, \sigma \rangle R \langle \partial(s), \sigma \rangle$ for all $i \geq 0$. Since Rule 21 is the final rule with $\langle t_i, \sigma \rangle \hookrightarrow \langle t_{i+1}, \sigma \rangle$ as conclusion, we have $r_0 = r$,

$$\langle r_0, \sigma \rangle \hookrightarrow \langle r_1, \sigma \rangle \hookrightarrow \langle r_2, \sigma \rangle \hookrightarrow \dots$$

and $t_i = \partial(r_i)$ for all $i \geq 0$. According to the definition of R , $\langle r_i, \sigma \rangle \sim_{isst} \langle s, \sigma \rangle$ for all $i \geq 0$ and therefore there exist s' and $j > 0$ such that $\langle s, \sigma \rangle \hookrightarrow \langle s', \sigma \rangle$ and $\langle r_j, \sigma \rangle \sim_{isst} \langle s', \sigma \rangle$. By applying Rule 21, $\langle \partial(s), \sigma \rangle \hookrightarrow \langle \partial(s'), \sigma \rangle$. According to the definition of R , $\langle \partial(r_j), \sigma \rangle R \langle \partial(s'), \sigma \rangle$.

With this we complete the proof that R is an interaction sensitive stuttering bisimulation relation.

5 Application

In [16], the translation from χ to PROMELA is discussed informally. It is pointed out that the translation is straightforward for some constructs of χ (e.g., for assignments and alternative composition), since they also exist in PROMELA. However, the translation of guards, nested scopes and nested parallelism is less straightforward, since they have no direct equivalents in PROMELA. In the preprocessing phase of the proposed translation, nested scopes and certain occurrences of nested parallelism are eliminated, and guards are pushed down to the level of atomic processes. In this section we indicate how this preprocessing phase can be proved correct using the notion of stuttering congruence proposed in this paper.

To be able to state some of the results pertaining to the scope operator, we first need to define the notion of *free* occurrence of a variable x in a process p .

Definition 13. *The set of free variables in $p \in P$ is defined as:*

$$free(p) = \begin{cases} \emptyset & \text{if } p = \varepsilon, \delta \text{ or skip} \\ \{x\} \cup \text{dom}(e) & \text{if } p = x := e \\ \text{dom}(e) & \text{if } p = m!e \\ \{x\} & \text{if } p = m?x \\ \text{dom}(b) \cup free(q) & \text{if } p = b \rightarrow q \\ free(q) \cup free(r) & \text{if } p = q \circ r, \text{ for all } \circ \in \{;, \parallel, \|\} \\ free(q) & \text{if } p = q^* \\ free(q) \setminus \text{dom}(s) & \text{if } p = \llbracket s \mid q \rrbracket \\ free(q) & \text{if } p = \partial(q) \end{cases}$$

The following theorem explains how scope operators can be pulled out.

Theorem 7. *Let $p, q \in P$, let $s, s_1, s_2 \in \Sigma$ and let $b \in B$ such that $\text{dom}(b) \cap \text{dom}(s) = free(p) \cap \text{dom}(s) = free(q) \cap \text{dom}(s) = \emptyset$. Then:*

1. $b \rightarrow \llbracket s \mid p \rrbracket \cong_{st} \llbracket s \mid b \rightarrow p \rrbracket$,
2. $\llbracket s \mid p \rrbracket ; q \cong_{st} \llbracket s \mid p ; q \rrbracket$,
3. $p ; \llbracket s \mid q \rrbracket \cong_{st} \llbracket s \mid p ; q \rrbracket$,
4. $\llbracket s \mid p \rrbracket \parallel q \cong_{st} \llbracket s \mid p \parallel q \rrbracket$,
5. $\llbracket s \mid p \rrbracket \|\! q \cong_{st} \llbracket s \mid p \|\! q \rrbracket$,
6. $\llbracket s_1 \mid \llbracket s_2 \mid p \rrbracket \rrbracket \cong_{st} \llbracket \gamma(s_1, s_2) \mid p \rrbracket$,
7. $\llbracket s \mid \partial(p) \rrbracket \cong_{st} \partial(\llbracket s \mid p \rrbracket)$.

In case of a repetition, we need to be careful: $\llbracket s \mid p \rrbracket^*$ is not stuttering congruent with $\llbracket s \mid p^* \rrbracket$, for in $\llbracket s \mid p \rrbracket^*$ the state s is applied before each repetition of p , while in $\llbracket s \mid p^* \rrbracket$ it is only applied before the first repetition. The solution is to incorporate in $\llbracket s \mid p^* \rrbracket$ the effect of the state s after each repetition of p by a sequential composition of assignments of the form $x := s(x)$, one for every $x \in \text{dom}(s)$. That is, if $\text{dom}(s) = \{x_1, \dots, x_n\}$, then we propose to replace $\llbracket s \mid p \rrbracket^*$ by $\llbracket s \mid (p ; x_1 := s(x_1) ; \dots ; x_n := s(x_n))^* \rrbracket$. Note, however, that this only works if p does not have the option to terminate immediately (i.e., if $\langle p, \sigma \rangle \not\downarrow$ for all states σ). For, if $\langle p, \gamma(s, \sigma) \rangle \downarrow$, then $\langle \llbracket s \mid (p ; x := c)^* \rrbracket, \sigma \rangle$ can do the action $aa(x, c)$, whereas $\langle \llbracket s \mid p \rrbracket^*, \sigma \rangle$ cannot do the same action (unless p can do it), and thus the root condition is violated.

Let $\overline{P}$ be the set of processes that do not contain ε and in which every occurrence of the star operator is immediately followed by the sequential composition operator. Then $\langle p, \sigma \rangle \not\downarrow$ for all $\sigma \in \Sigma$ and all $p \in \overline{P}$, as is easily proved by structural induction on p .

Theorem 8. *Let $p \in \overline{P}$, let $x \in V$, and let $c \in C$. If s is a state such that $\text{dom}(s) = \{x_1, \dots, x_n\}$, then*

$$\llbracket s \mid p \rrbracket^* \cong_{st} \llbracket s \mid (p ; x_1 := s(x_1) ; \dots ; x_n := s(x_n))^* \rrbracket.$$

The next theorem explains how guards can be distributed over all operators except parallel composition.

Theorem 9. *Let $p, q \in P$, let $s \in \Sigma$ and let $b, b_1, b_2 \in B$, then:*

1. $true : \rightarrow p \cong_{st} p$
2. $b_1 : \rightarrow b_2 : \rightarrow p \cong_{st} (b_1 \wedge b_2) : \rightarrow p$
3. $b : \rightarrow (p ; q) \cong_{st} (b : \rightarrow p) ; q$
4. $b : \rightarrow (p \parallel q) \cong_{st} (b : \rightarrow p) \parallel (b : \rightarrow q)$
5. $b : \rightarrow p^* \cong_{st} (b : \rightarrow p) ; p^* \parallel b : \rightarrow \varepsilon$
6. $b : \rightarrow \partial(p) \cong_{st} \partial(b : \rightarrow p)$

Note that the process $b : \rightarrow (p \parallel q)$ is not stuttering congruent with the process $(b : \rightarrow p) \parallel (b : \rightarrow q)$. For suppose the processes are executed in a state in which b is *true* and an action from p changes the state in such a way that the value of b becomes *false*. Then the process $b : \rightarrow (p \parallel q)$ proceeds as $p' \parallel q$ and the process $(b : \rightarrow p) \parallel (b : \rightarrow q)$ as $p' \parallel (b : \rightarrow q)$, and in the latter process the option q is blocked. We finish this section with a theorem that allows us to simplify nested parallelism in the context of sequential composition. For similar reasons as before it is formulated in terms of the set $\overline{P}$.

Theorem 10. *Let $p, q, r \in \overline{P}$, and let $w \in V$ such that $w \notin \text{free}(p) \cup \text{free}(q) \cup \text{free}(r)$. Then,*

1. $(p \parallel q) ; r \cong_{st} \llbracket w \mapsto 0 \mid p ; w := w + 1 \parallel q ; w := w + 1 \parallel (w = 2) : \rightarrow r \rrbracket,$
2. $p ; (q \parallel r) \cong_{st} \llbracket w \mapsto 0 \mid p ; w := w + 1 \parallel (w = 1) : \rightarrow q \parallel (w = 1) : \rightarrow r \rrbracket.$

6 Conclusion

In this paper we have proposed the notion of stuttering congruence for the modeling and simulation language χ . We have proved that if two χ processes are stuttering congruent, then, with respect to any state σ , they satisfy the same CTL^*_{-X} formulas and either both have deadlock or neither of them. Stuttering congruence is a behavioural congruence for the constructs of discrete-event, untimed part of χ , i.e., it is defined directly on the operational semantics of the language and it is a congruence for its constructions. Therefore, it is suitable for establishing the correctness of syntactic transformations on χ models.

We have illustrated the use of stuttering congruence in correctness proofs of syntactic transformations by indicating how (a part of) the preprocessing phase of the translation from χ to PROMELA can be proved correct. It is explained in detail in [16] that if a χ model satisfies a few general restrictions, then the preprocessing phase yields a χ model that can be straightforwardly translated into PROMELA. The resulting PROMELA specification can then be verified with SPIN. Incidentally, note that for Theorems 8 and 10 it is essential that stuttering congruence allows ‘stuttering’; the transformations in these theorem do not preserve the validity of full CTL^* .

Currently, a translation from χ to UPPAAL is being developed, and it also involves a preprocessing phase. The model checker of UPPAAL verifies CTL formulas, and since our stuttering congruence preserves CTL^*_{-X} formulas, it will be suitable for proving the correctness of that preprocessing phase too.

As future work, we mention the extension of the results obtained in this paper to full timed χ [15], proving the preservation of the validity of a timed variant of CTL^*_{-X} .

References

1. S. Blom, W. Fokkink, J.F. Groote, I. van Langevelde, B. Lisser, and J.C. van de Pol. μ CRL: A toolset for analysing algebraic specifications. In *Proceedings of CAV2001*, LNCS 2102, pages 250–254, 2001.
2. E. Bortnik, N. Trčka, A.J. Wijs, S.P. Luttik, J.M. van de Mortel-Fronczak, J.C.M. Baeten, W.J. Fokkink, and J.E. Rooda. Analyzing a χ model of a turntable system using SPIN, CADP and UPPAAL. *Journal of Logic and Algebraic Programming*, 2005. To appear.
3. V. Bos and J.J.T. Kleijn. Automatic verification of a manufacturing system. *Robotics and Computer Integrated Manufacturing*, 17:185–198, 2001.
4. V. Bos and J.J.T. Klein. *Formal specification and analysis of industrial systems*. PhD thesis, Eindhoven University of Technology, 2002.
5. M.C. Browne, E.M. Clarke, and O. Grumberg. Characterizing finite Kripke structures in propositional temporal logic. *Theor. Comput. Sci.*, 59:115–131, 1988.
6. E.J.J. van Campen. *Design of a multi-process multi-product wafer fab*. PhD thesis, Eindhoven University of Technology, 2000.
7. E.M. Clarke, O. Grumberg, and D.A. Peled. *Model checking*. MIT Press, Cambridge, Massachusetts, 1999.
8. J.-C. Fernandez, H. Garavel, A. Kerbrat, L. Mounier, R. Mateescu, and M. Sighireanu. CADP - a protocol validation and verification toolbox. In *Proceedings 8th of CAV'96*, LNCS 1102, pages 437–440, 1996.
9. J.J.H. Fey. *Design of a fruit juice blending and packaging plant*. PhD thesis, Eindhoven University of Technology, 2000.
10. R. Gerth, R. Kuiper, D. Peled, and W. Penczek. A partial order approach to branching time logic model checking. *Information and Computation*, 150(2):132–152, 1999.
11. J. A. Govaarts. Efficiency in a lean assembly line: a case study at NedCar born. Master Thesis, 1997.
12. G.J. Holzmann. *The SPIN model checker*. Addison-Wesley, 2003.
13. K.G. Larsen, P. Pettersson, and W. Yi. UPPAAL in a nutshell. *Int. Journal on Software Tools for Technology Transfer*, 1(1-2):134–152, 1997.
14. R. De Nicola and F.W. Vaandrager. Three logics for branching bisimulation. *J. ACM*, 42(2):458–487, 1995.
15. R.R.H. Schiffelers, D.A. van Beek, K.L. Man, M.A. Reniers, and J.E. Rooda. Syntax and semantics of timed Chi. Computer Science Report 05/09, Eindhoven University of Technology, 2005.
16. N. Trčka. Verifying χ models of industrial systems with Spin. Computer Science Report 05/12, Eindhoven University of Technology, 2005.
17. D.A. van Beek, A. van der Ham, and J.E. Rooda. Modelling and control of process industry batch production systems. In *15th Triennial World Congress of the International Federation of Automatic Control*, Barcelona, Spain, CD-ROM, 2002.