Congruence Formats for Weak Readiness Equivalence and Weak Possible Future Equivalence *

Xiaowei Huang[1], Li Jiao[2], Weiming Lu[1]
[1]Academy of Mathematics and System Science,
Chinese Academy of Sciences.
[2]State Key Laboratory of Computer Science,
Institute of Software, Chinese Academy of Sciences.

Abstract

Weak equivalences are important behavioral equivalences in the course of specifying and analyzing reactive systems using process algebraic languages. In this paper, we propose a series of weak equivalences named weak parametric readiness equivalences, which take two previously-known behavioral equivalences, i.e., the weak readiness equivalence and the weak possible future equivalence, as their special cases. More importantly, based on the idea of Structural Operational Semantics, a series of rule formats are further presented to guarantee congruence for these weak parametric readiness equivalences, i.e., to show that the proposed rule formats can guarantee the congruence of their corresponding weak parametric readiness equivalences. This series of rule formats reflect the differences in the weak parametric readiness equivalences. We conclude that, when the weak parametric readiness equivalences become coarser, their corresponding rule formats turn tighter.

1 Introduction

Behavioral equivalence is based on observability and thus equivalences may differ by the notions of observability [1, 2]. A natural classification of behavioral equivalence is that a given behavioral equivalence may be strong or weak. Their difference stems from the ways of dealing with the internal transitions, which are generally denoted as τ transitions. Strong equivalences regard τ transitions as the observable actions. Weak equivalences, on the other hand, suppose them unobserved by the outer-world. In this paper, we will focus on the weak equivalences.

Among various weak semantic equivalences, the weak readiness equivalence and the weak possible future equivalence are two interesting semantic equivalences. The

*This research was financially supported by the National Natural Science Foundation of China (No. 60421001).

readiness equivalence, which is the strong counterpart of weak readiness equivalence, was proposed by Olderog and Hoare [3], and possible future semantics, which is the strong counterpart of weak possible future equivalence, was proposed by Rounds and Brooks [4]. Besides their previous applications as important semantics for communicating processes, Ryan [5] has explored the application of weak readiness semantics in computer security by claiming that it is more intuitive to think it in terms of the weak readiness equivalence than in weak failure equivalence. However, as will be shown in the paper, weak readiness equivalence is not congruent on the hiding operator of CSP [6]. On the other hand, weak possible future equivalence is congruent on the hiding operator. Therefore, we can anticipate the use of weak possible future equivalence in computer security.

Technically speaking, weak possible future equivalence is strictly finer than weak readiness equivalence. The weak readiness semantics is usually denoted, by its denotational characterization, as a set of weak readiness pairs. Two processes are weak readiness equivalent iff they have the same sets of weak readiness pairs. Likewise, the weak possible future semantics is usually denoted as a set of weak possible future pairs and two processes are weak possible future equivalent iff they have the same sets of weak possible future pairs.

Looking into the two pairs, we find that their first parameters both express the abilities: some process p executes a sequence of observable actions and evolves into another process p'. The difference exists in their second parameters. The second parameter of a weak readiness pair is the set of actions enabled by p', and the second parameter of a weak possible future pair is the set of action sequences enabled by p'.

Based on these observations, we define a series of weak equivalences, called weak parametric readiness equivalences. Like the above two weak equivalences, a weak i-readiness pair with $i \in \mathbb{N} \cup \{\omega\}$ is only different in its second parameter from the weak readiness pair, where $\mathbb{N}$ is the set of natural numbers and ω is the cardinality of $\mathbb{N}$. Its second parameter is the set of action sequences enabled by p' and the lengths of these action sequences do not exceed i. Therefore, the plain weak readiness equivalence is the weak 1-readiness equivalence in our framework and the weak possible future equivalence is the weak ω-readiness equivalence. Furthermore, with the increasing of parameter i, the weak i-readiness equivalence becomes finer.

Structural Operational Semantics (SOS) [7] has been widely used in defining the meanings of the operators in various process algebraic languages, such as CCS [8] and ACP [9]. The main idea of SOS is: each process is represented by a closed term and has some out-going transitions to denote its behaviors. Each transition shows an action which the process can take and then evolves into another process. Then, to each n-ary syntactic symbol, a number of transition rules are assigned. These transition rules specify how to generate the outgoing-transitions of $f(x_1, ..., x_n)$ via the outgoing-transitions of its arguments $x_1, ..., x_n$. Therefore, the behavior of $f(p_1, ..., p_n)$ can be deduced from the behaviors of $p_1, ..., p_n$ and the transition rules of f.

Transition System Specifications (TSS's) [10], which are borrowed from logic programming, form a theoretical basis for SOS. By imposing some syntactic restrictions on TSS's, one can retrieve so-called rule formats. From a specified rule format, one may deduce some interesting properties. Among these properties, one of the most important ones is whether or not a behavioral equivalence is congruent for a TSS in this rule format. As stated in [11], an equivalence relation $\sim$ is congruent on some TSS, if $\sim$ satisfies the compatibility property, which states that, for any n-ary function symbol f in the TSS and processes p_i, q_i, if $p_i \sim q_i$ for $1 \leq i \leq n$ then $f(p_1, ..., p_n) \sim f(q_1, ..., q_n)$. In the rest of the paper, a TSS is always called a language.

Up to now, some rule formats have been presented to meet the behavioral equivalences, for examples, GSOS format [12] and ntyft/nxyft format [13] have been proved to be congruent on strong bisimulation, de Simone [14] format was proved to be congruent on failure equivalence, and so on. More works have been done on pursuing a suitable rule format for a given strong equivalence. On the other hand, much less attention was paid on the rule formats for weak equivalences. More specifically, to our knowledge, no congruence formats have been presented for the weak readiness equivalence or the weak possible future equivalence. Aceto, Fokkink and Verhoef [16] and Mousavi, Reniers and Groote [17] have provided comprehensive overviews on the rule formats.

In the paper, we will propose a series of rule formats for the newly defined weak parametric readiness equivalences. In fact, weak 1-readiness format is presented for the weak 1-readiness equivalence, weak finite readiness format for the weak i-readiness equivalences with $1 < i < \omega$, and weak ω-readiness format for the weak ω-readiness equivalence. Then, we prove that the weak parametric readiness equivalence can be preserved after composition if the language is in its corresponding rule formats, i.e., these rule formats are all congruence formats for their corresponding equivalences.

Here, we want to sketch out two critical points in pursuing these rule formats:

The first critical point is on the feasibility of allowing the rules with τ-conclusion. Rules with τ-conclusion are an important class of rules in classical process algebraic languages, notable examples include a transition rule of the hiding operator of CSP and a transition rule of the parallel composition operator of CCS.

$$\frac{x \xrightarrow{a} x'}{x/A \xrightarrow{\tau} x'/A} a \in A \tag{1}$$

$$\frac{x \xrightarrow{a} x', y \xrightarrow{b} y'}{x|y \xrightarrow{\tau} x'|y'} (a, b) \in f \tag{2}$$

However, not all behavioral equivalences can be preserved under these rules, such as, the acceptance testing equivalence may not be preserved under the hiding operator [15]. In this paper, we will take a close look into these rules. In fact, the weak i-readiness equivalences with $i < \omega$ may not be preserved under these rules, but the weak ω-readiness equivalence, i.e., the weak possible future equivalence, can survive these rules.

The second critical point is whether or not the patience rules for receiving arguments are necessary in the rule formats for a given weak parametric readiness equivalence. Patience rules, which are used to smooth the evolvement of τ transitions of subprocesses, are usually necessary in rule formats for weak equivalences. For example, assume process $a|\tau b$ and the other two transition rules of parallel composition operator besides expression (2).

$$\frac{p \xrightarrow{a} p'}{p|q \xrightarrow{a} p'|q}, \frac{q \xrightarrow{a} q'}{p|q \xrightarrow{a} p|q'} \tag{3}$$

Patience rule $\frac{q \xrightarrow{\tau} q'}{p|q \xrightarrow{\tau} p|q'}$ will be applied before the communication of subprocesses a and b. However, since patience rules are defined in accordance with the arguments of an operator, they can be divided into three classes: patience rules for active arguments, patience rules for receiving arguments and patience rules for other arguments [19, 20]. Though patience rules for active arguments are generally needed, we find that, patience rules for receiving arguments are not necessary for some rule formats. For example, for the weak 1-readiness format, patience rules for receiving arguments are not necessary because of the exclusion of rules with τ-conclusion. On the other hand, they are necessary for the weak i-readiness format with $i > 1$.

As a result, the weak finite readiness format is tighter than the weak ω-readiness format because the rules with τ-conclusion should be excluded from the language in the weak finite readiness format, and the weak 1-readiness format is tighter than the weak finite readiness format since it may further exclude the patience rules for receiving arguments from the language. Therefore, we can conclude that, when the weak parametric readiness equivalences become coarser, their corresponding rule formats turn tighter.

Finally, we want to say more on the newly-proposed weak i-readiness equivalences with $1 < i < n$. We have not found their precise applications, though they can be used in most applications of the 1-readiness equivalences. The reasons that we introduce these intermediate weak equivalences are that

1) they can smooth the changes between the weak 1-readiness equivalence and the weak ω-readiness equivalence,

2) their congruence format is also an intermediate format between the weak 1-readiness format and the weak ω-readiness format, and

3) most importantly, we want to make clear the technical reasons why there exist differences between the weak 1-readiness format and the weak ω-readiness format. Take it more concrete, from the weak ω-readiness format to the weak 1-readiness format, the reason why the rules with τ conclusion are excluded is that the parameter i degrades from infinite to finite, and the reason why the patience rules for receiving arguments are not necessary is that, no matter how they are presented in the language, only the set of next one, but not next finite or infinite, observable actions remains unique.

The structure of this paper is: in Section 2, we will introduce some preliminaries, mainly on the behavioral equivalences and the rule formats in Structural Operational Semantics. Then in Section 3, we will put forward the formal definitions of the weak parametric readiness equivalences. Intuitive motivations on their rule formats will be illustrated by examples in Section 4. Section 5 is devoted to the formal definitions of the three congruence formats, and the proofs on the congruence theorems. And then, in Section 6, we will conclude the paper.

2 Preliminaries on Behavior Equivalences and Rule Formats

Let Act denote a set of names which will be used to label events and Act^* be the set of all action sequences. We usually use $a, b, ...$ to range over the actions in Act, and use $A, B, ...$ to range over subsets of actions in Act. τ is generally used to denote the internal action which can not be observed by the outer world, and we use $\alpha, \beta, ...$ to range over the actions in $Act \cup \{\tau\}$. $\delta, \mu, \sigma, ...$ is to range over the sequences of actions. $\Phi, \Psi, ...$ is to range over the sets of sequences. $p, q, ...$ will be used to represent processes.

Any behavioral semantics of some process p can be characterized by a function $O(p)$ [1]. $O(p)$ constitutes the observable behaviors of p. The equivalence relation $\sim_O$ can be defined by $p \sim_O q \Longleftrightarrow O(p) = O(q)$. The readers are referred to van Glabbeek [1, 2] for comprehensive reviews of the behavioral equivalences.

SOS has been widely accepted as a tool to define operational semantics of processes. A TSS is a formalization of SOS [7]. The readers are referred to Aceto, Fokkink and Verhoef [16] for a comprehensive review on SOS.

Definition 2.1 [16] Let $V = \{x_1, x_2, ...\}$ be a set of variables. A signature Σ is a collection of function symbols $f \notin V$ equipped with a function $ar : \Sigma \rightarrow N$. The set $\mathbb{T}(\Sigma)$ of terms over a signature Σ is defined recursively by: 1) $V \subseteq \mathbb{T}(\Sigma)$; 2) if $f \in \Sigma$ and $t_1, ..., t_{ar(f)} \in \mathbb{T}(\Sigma)$, then $f(t_1, ..., t_{ar(f)}) \in \mathbb{T}(\Sigma)$.

A term $c()$ is abbreviated as c. For $t \in \mathbb{T}(\Sigma)$, $var(t)$ denotes the set of variables that occur in t. $\mathrm{T}(\Sigma)$ is the set of closed terms over Σ, i.e., the terms $p \in \mathbb{T}(\Sigma)$ with $var(p) = \emptyset$. A Σ substitution ζ is a mapping from V to $\mathbb{T}(\Sigma)$, and therefore, a closed Σ substitution ζ is a mapping from V to $\mathrm{T}(\Sigma)$.

Definition 2.2 A positive Σ-literal is an expression $t \xrightarrow{\alpha} t'$ and a negative Σ-literal is an expression $t \overset{\alpha}{\nrightarrow}$ with $t, t' \in \mathbb{T}(\Sigma)$ and $\alpha \in Act \cup \{\tau\}$. A transition rule over Σ is an expression of the form $\frac{H}{C}$ with H a set of Σ literals (the premises of the rule) and C a positive Σ-literal (the conclusion). The left- and right-hand side of C are called the source and the target of the rule, respectively. Moreover, if $r = \dfrac{H}{t \xrightarrow{\alpha} t'}$ then define $ante(r) = H$, $cons(r) = \{t \xrightarrow{\alpha} t'\}$, and the output of r as α.

A TSS, written as (Σ, Ψ), consists of a signature Σ and a set Ψ of transition rules over Σ. A TSS is positive if the premises of its rules are positive.

Definition 2.3 Let Σ be a signature. A context C of n holes over Σ is simply a term in $\mathbb{T}(\Sigma)$ in which n variables occur, each variable only once. If $t_1, ..., t_n$ are terms over Σ, then $C(t_1, ..., t_n)$ denotes the term obtained by substituting t_1 for the first variable occurring in C, t_2 for the second variable occurring, etc. Thus, if $x_1, ..., x_n$ are all different variables, then $C(x_1, ..., x_n)$ denotes a context of n holes in which x_i is the ith occurring variable.

Then, we can give the definition on the congruence of an equivalence in a language.

Definition 2.4 Let $\mathcal{L} = (\Sigma, \Psi)$ be a language. An equivalence relation $\sim$ is congruent on language $\mathcal{L}$ iff $\forall i \in \{1, ..., n\} : p_i \sim q_i \Longrightarrow C(p_1, ..., p_n) \sim C(q_1, ..., q_n)$ for any context C of n holes in language $\mathcal{L}$, where p_i and q_i are closed terms, i.e., processes, over Σ.

Definition 2.5 Let Σ be a signature. A transition relation over Σ is a relation Tr

$\subseteq \mathrm{T}(\Sigma) \times (Act \cup \{\tau\}) \times \mathrm{T}(\Sigma)$. Element (p, α, p') of a transition relation is written as $p \xrightarrow{\alpha} p'$.

Thus a transition relation over Σ can be regarded as a set of closed positive Σ-literals(transitions).

Furthermore, for an action sequence $\delta = \alpha_1 ... \alpha_n$, if there exist $p_1, ..., p_n \in \mathrm{T}(\Sigma)$ such that $p \xrightarrow{\alpha_1} p_1 \xrightarrow{\alpha_2} ... \xrightarrow{\alpha_n} p_n$, then we call δ a trace of p, denoted as $p \xrightarrow{\delta}$ or $p \xrightarrow{\alpha_1} ... \xrightarrow{\alpha_n}$.

In weak semantics, the weak transition relations and the weak traces are also needed to be defined. Let p be a process, we write $p \stackrel{a}{\Longrightarrow}$ iff $p \xrightarrow{\tau^*}\xrightarrow{a}\xrightarrow{\tau^*}$, where τ^* denotes any number of internal transitions. Hence, for an observable action sequence $\delta = a_1 ... a_n$, $p \stackrel{\delta}{\Longrightarrow}$ iff $p \stackrel{a_1}{\Longrightarrow} ... \stackrel{a_n}{\Longrightarrow}$.

By imposing some syntactic constraints on TSS's, we will obtain the so-called rule formats with some properties on their induced operational semantics. Within these properties, it is specially important that whether a behavioral equivalence can be preserved in the languages with this format. Some rule formats have been proposed to meet the numerous behavioral equivalences, such as GSOS format, de Simone format, ntyft/nxyft format, etc. The readers are referred to Mousavi, Reniers and Groote [17] for the latest review on the rule formats.

The de Simone language will be employed as our starting point in retrieving the rule formats for the weak parametric readiness equivalences.

Definition 2.6 [14] Let Σ be a signature. A transition rule r is in de Simone format if it has the form $\dfrac{\{x_i \xrightarrow{a_i} y_i\}_{i \in I}}{f(x_1, ..., x_{ar(f)}) \xrightarrow{a} t}$, where $I \subseteq \{1, ..., ar(f)\}$ and the variables x_i and y_i are all distinct and the only variables occuring in r. Moreover, the target $t \in \mathbb{T}(\Sigma)$ does not contain variable x_i for $i \in I$ and has no multiple occurrence of variables.

Moreover, a language $\mathcal{L} = (\Sigma, \Psi)$ is in de Simone format if all transition rules in Ψ are in de Simone format. Besides, we also call $\mathcal{L} = (\Sigma, \Psi)$ a de Simone language if it is in de Simone format.

Below, two special classes of rules are defined. They will be discussed in the paper. The first class is the patience rules, and the second class is the rules with τ-conclusion.

Definition 2.7 [18, 19] Let $\mathcal{L} = (\Sigma, \Psi)$ be a de Simone language, and f be a function symbol in Σ. A rule of the form $\dfrac{x_i \xrightarrow{\tau} x_i'}{f(x_1, ..., x_i, ..., x_n) \xrightarrow{\tau} f(x_1, ..., x_i', ..., x_n)}$ with $1 \leq i \leq n$ is called a patience rule of the ith argument of f.

In the following, a rule is called a plain rule if it is not a patience rule.

Definition 2.8 [20] Let $\mathcal{L} = (\Sigma, \Psi)$ be a de Simone language, and f be a function symbol in Σ. An argument $i \in N$ of an operator f is active if f has a rule in which x_i appears as left-hand side of a premise. A variable x occurring in a term t is receiving in t if t is the target of a rule in which x is the right-hand side of a premise. An argument $i \in N$ of an operator f is receiving if a variable x is receiving in a term t that has a subterm $f(t_1, ..., t_n)$ with x occurring in t_i.

Then, the set of all arguments Arg of an operator can be divided into three classes: active arguments Arg_a, receiving arguments Arg_r and others Arg_o, which is inspired by van Glabbeek [20]. Therefore, $Arg = Arg_a + Arg_r + Arg_o$.

Similarly, patience rules of an operator can be divided into three classes. It should be noted that an argument may be both an active argument and a receiving argument, i.e., $Arg_a \cap Arg_r \neq \phi$. However for clarity, from now on, if we say that an argument is a receiving argument, then it should not be an active argument , i.e., receiving arguments below are only those receiving arguments which are not active arguments simultaneously. Therefore, Arg_a, Arg_r and Arg_o will be disjoint.

Definition 2.9 Let $\mathcal{L} = (\Sigma, \Psi)$ be a de Simone language, and f be a function symbol in Σ. A rule of the form $\dfrac{H}{f(x_1, ..., x_n) \xrightarrow{\tau} t}$ is called a rule with τ-conclusion, if it is not a patience rule and there exists at least one positive Σ literal in H.

A notable example of the rules with τ-conclusion, which will be used in Section 4.2, is the first transition rule of the hiding operator in CSP as follows.

$$p/A : \frac{p \xrightarrow{\alpha} p'}{p/A \xrightarrow{\tau} p'/A}\ \alpha \in A \quad \frac{p \xrightarrow{\alpha} p'}{p/A \xrightarrow{\alpha} p'/A}\ \alpha \notin A$$

Like the definition of a rule with τ-conclusion, a transition rule is a rule with τ-premise iff there exists a positive Σ literal like $t \xrightarrow{\tau} t'$ in its premises. It is trivial that patience rules are rules with τ-premise.

Before concluding this section, we will presume a small set of operators with default operational semantics:

nil : means successful termination.

$a \cdot X : a \cdot X \xrightarrow{a} X$

$$X \boxplus Y : \frac{X \xrightarrow{a} X'}{X \boxplus Y \xrightarrow{a} X'} \quad \frac{Y \xrightarrow{a} Y'}{X \boxplus Y \xrightarrow{a} Y'} \quad \frac{X \xrightarrow{\tau} X'}{X \boxplus Y \xrightarrow{\tau} X' \boxplus Y} \quad \frac{Y \xrightarrow{\tau} Y'}{X \boxplus Y \xrightarrow{\tau} X \boxplus Y'}$$

$X \oplus Y : X \oplus Y \xrightarrow{\tau} X \quad X \oplus Y \xrightarrow{\tau} Y$

$$X \rhd Y : \frac{X \xrightarrow{a} X'}{X \rhd Y \xrightarrow{a} X'} \quad \frac{X \xrightarrow{\tau} X'}{X \rhd Y \xrightarrow{\tau} X' \rhd Y} \quad X \rhd Y \xrightarrow{\tau} Y$$

where $a \in Act$. We call this language **B** [21].

3 Weak Parametric Readiness Equivalences

Before presenting the formal definitions of the weak parametric readiness equivalences, two canonical equivalences, i.e., the weak readiness equivalence and the weak possible future equivalence, will be introduced. As we will see, they both are the special cases of the weak parametric readiness equivalences.

Definition 3.1 $(\sigma, A) \in Act^* \times \mathcal{P}(Act)$ is a weak readiness pair of process p iff there exists some p' such that $p \overset{\sigma}{\Longrightarrow} p' \wedge A = \mathcal{S}(p')$, where $\mathcal{S}(p') = \{a \in Act \mid p' \overset{a}{\Longrightarrow}\}$. The set of all weak readiness pairs of process p is called the weak readiness of p, denoted as $\mathcal{R}(p)$.

Weak Readiness Equivalence $\sim_r$: for any two processes p and q, $p \sim_r q$ iff $\mathcal{R}(p) = \mathcal{R}(q)$.

Definition 3.2 $(\sigma, \Phi) \in Act^* \times \mathcal{P}(Act^*)$ is a weak possible future pair of process p iff there exists some p' such that $p \overset{\sigma}{\Longrightarrow} p' \wedge \Phi = \mathcal{T}(p')$, where $\mathcal{T}(p') = \{\delta \in Act^* \mid p' \overset{\delta}{\Longrightarrow}\}$. The set of all weak possible future pairs of process p is called the weak possible future of p, denoted as $\mathcal{PF}(p)$.

Weak Possible Future Equivalence $\sim_{pf}$: p and q are two processes, $p \sim_{pf} q$ iff $\mathcal{PF}(p) = \mathcal{PF}(q)$.

As can be seen in the above definitions, the difference between the weak readiness pair and the weak possible future pair exists on their second parameters. The second parameter of the weak readiness pair is a set of actions enabled by p'. From this respect, we may also say in the paper that weak readiness semantics sees an observable action ahead. On the other hand, the second parameter of the weak possible future pair is a set of action sequences enabled by p' and hence it is said that the weak possible future semantics sees ω observable actions ahead.

By the above observation, we put forward the definition on the weak parametric readiness equivalences:

Definition 3.3 $(\sigma, \Phi) \in Act^* \times \mathcal{P}(Act^*)$ is a weak i-readiness pair of process p iff there exists some p' such that $p \overset{\sigma}{\Longrightarrow} p' \wedge \Phi = \mathcal{T}(p', i)$, where $\mathcal{T}(p', i) = \{\delta \in Act^* \mid p' \overset{\delta}{\Longrightarrow} \wedge |\delta| \leqslant i\}$. The set of all weak i-readiness pair of process p is called the weak i-readiness of p, denoted as $\mathcal{R}(p, i)$.

Weak Parametric Readiness Equivalences $\sim_r^i$: for any two processes p and q, $p \sim_r^i q$ iff $\mathcal{R}(p, i) = \mathcal{R}(q, i)$.

Also, we will often say that p and q are weak i-readiness equivalent if $p \sim_r^i q$.

It is trivial that, in this framework, the weak readiness equivalence is the weak 1-readiness equivalence and the weak possible future equivalence is the weak ω-readiness equivalence. In fact, $\mathcal{S}(p') = \mathcal{T}(p', 1)$ and $\mathcal{T}(p') = \mathcal{T}(p', \omega)$.

The theorem below states that if p and q are weak j-readiness equivalent with $1 \leq j \leq \omega$, then they are also weak i-readiness equivalent for $i < j$.

Theorem 3.4 Let $1 \leq i < j \leq \omega$, p and q are two processes. If $p \sim_r^j q$ then $p \sim_r^i q$.

Proof By the definition of weak parametric readiness equivalences, $p \sim_r^j q$ iff $\mathcal{R}(p, j) = \mathcal{R}(q, j)$. Then, $\mathcal{R}(p, i) = \mathcal{R}(q, i)$ can be obtained from Definition 3.3, $\mathcal{R}(p, j) = \mathcal{R}(q, j)$ and $1 \leq i < j \leq \omega$. Therefore, $p \sim_r^i q$. $\square$

4 Intuitive Motivations on Rule Formats

This section gives several representative examples to show some intuitive motivations on the rule formats of the weak parametric readiness equivalences. However, we do not want to discuss them from the scratch, only the two critical points sketched in the introduction are to be mentioned: the first subsection is to observe the feasibility of adding rules with τ-conclusion; the second subsection is to inspect the necessity of the patience rules for receiving arguments. Besides, in the third subsection, we will make a discussion by means of a simple example on the infeasibility of introducing the negative premises into a rule, when confronted with a weak equivalence.

It should be noted that, in this section, we are mainly concerned the intuitive motivations. The results retrieved in this section will be formally defined and proved in the next section. Also, as the starting point, we assume the basic language **B** which has been introduced in section 2.

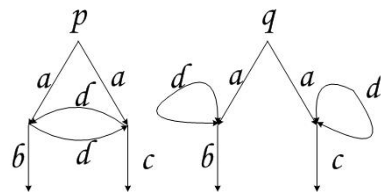

Figure 1: p and q are weak 1-readiness equivalent, but $p/\{d\}$ and $q/\{d\}$ are not weak 1-readiness equivalent.

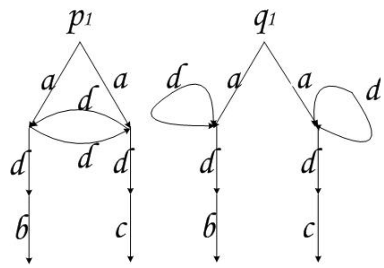

Figure 2: p_1 and q_1 are weak 2-readiness equivalent, but $p_1/\{d\}$ and $q_1/\{d\}$ are not weak 2-readiness equivalent.

4.1 On Rules with τ-conclusion

Let's consider the example in Figure 1 and Figure 2. Firstly, the two graphs in Figure 1, i.e., p and q, are weak 1-readiness equivalent. However, after hiding d actions, $p/\{d\}$ is not weak 1-readiness equivalent to $q/\{d\}$, which can be seen from the weak 1-readiness pair $(a, \{b, c\}) \in \mathcal{R}(p/\{d\}, 1)$ but $(a, \{b, c\}) \notin \mathcal{R}(q/\{d\}, 1)$. If we take the weak 2-readiness equivalence into consideration, we will find that p and q are not yet weak 2-readiness equivalent.

As for the weak 2-readiness equivalence, p_1 and q_1, the two graphs in Figure 2, are weak 2-readiness equivalent. However, this equivalence also cannot be preserved after hiding d actions. In fact, for any weak i-readiness equivalence with $i < \omega$, a similar counterexample exists. On the contrary, the weak ω-readiness equivalence can be preserved under the hiding operator.

Generalizing to all rules with τ-conclusion, a common characterization of these rules is that they all consume the observable actions of the subprocesses and produce τ transitions at the same time. Therefore, we conjecture that any weak i-readiness equivalence with $i < \omega$ will probably be broken under the rules with τ-conclusion, but the weak ω-readiness equivalence will be preserved.

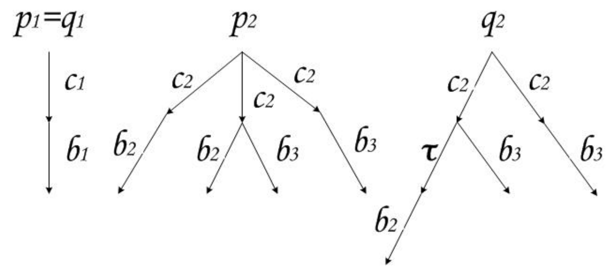

Figure 3: p_1 and q_1, p_2 and q_2 are weak i-readiness equivalent for $i \in \mathbb{N} \cup \{\omega\}$.

4.2 Patience Rules for Receiving Arguments

Consider adding the rules $r_1 = \dfrac{x_1 \xrightarrow{c_1} x_1', x_2 \xrightarrow{c_2} x_2'}{f(x_1, x_2) \xrightarrow{a_1} g(x_1', x_2')}$, $r_2 = \dfrac{x_1 \xrightarrow{b_1} x_1'}{g(x_1, x_2) \xrightarrow{a_2} h(x_2)}$, $r_3 = \dfrac{x_1 \xrightarrow{b_3} x_1'}{h(x_1) \xrightarrow{b_3} nil}$, $r_4 = \dfrac{x_1 \xrightarrow{b_2} x_1'}{h(x_1) \xrightarrow{b_2} nil}$ and their associated patience rules for active arguments into language **B**.

By Definition 2.8, to decide the division of some argument of a given operator f, we need to take into consideration all transition rules in which operator f appears.

Definition 2.8 says that, if f has a rule in which x_i appears as left-hand side of a premise then x_i is an active argument. For the case of operator $g(x_1, x_2)$ as above, x_1 is an active argument since it appears as left-hand side of the premise of rule r_2. On the other hand, x_2 is not an active argument, because no transition rule of operator $g(x_1, x_2)$ has x_2 as left-hand side of one of its premises.

Then, x_1 and x_2 are both receiving arguments by rule r_1. Therefore, x_1 is both active and receiving, and x_2 is a receiving argument.

Now, by the statement after Definition 2.8, an argument will be active if it is both active and receiving. Therefore, x_1 should be classified as active argument.

In all, for the case of operator $g(x_1, x_2)$ as above, x_1 is an active argument and x_2 is a receiving argument.

Let p_1, p_2, q_1 and q_2 be the processes shown in Figure 3. It can be easily verified that $p_1 \sim_r^1 q_1$ and $p_2 \sim_r^1 q_2$. According to the above rules, we have $f(p_1, p_2)$ and $f(q_1, q_2)$ shown in the two graphs of Figure 4. Now, $f(p_1, p_2)$ and $f(q_1, q_2)$ are also weak 1-readiness equivalent, i.e., $f(p_1, p_2) \sim_r^1 f(q_1, q_2)$. Therefore, it seems that patience rules for receiving arguments are not necessary for weak 1-readiness equivalence.

For the weak 2-readiness equivalence, we also have $p_1 \sim_r^2 q_1$ and $p_2 \sim_r^2 q_2$. $f(p_1, p_2)$ and $f(q_1, q_2)$ are not weak 2-readiness equivalent yet, because $(a_1, \{a_2 b_2\})$ is a weak 2-readiness pair of $f(p_1, p_2)$ but not a weak 2-readiness pair of $f(q_1, q_2)$. Hence, the weak 2-readiness equivalence is not preserved under the above rules. However, if we further add patience rule for the second argument of $g(x_1, x_2)$ into language

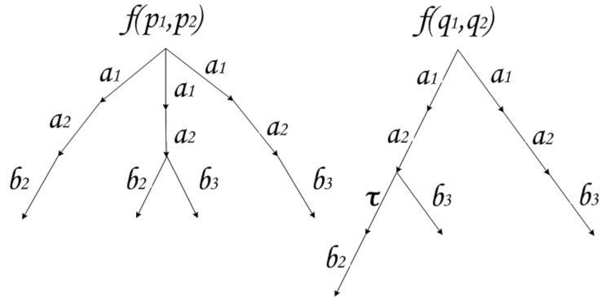

Figure 4: $f(p_1, p_2)$ and $f(q_1, q_2)$ are weak 1-readiness equivalent but not weak 2-readiness equivalent.

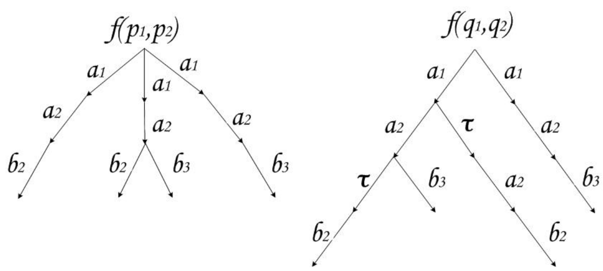

Figure 5: $f(p_1, p_2)$ and $f(q_2, q_2)$ are weak 2-readiness equivalent.

B, then $f(p_1, p_2)$ and $f(q_1, q_2)$ are the two graphs in Figure 5. Now, it can be easily verified that $f(p_1, p_2) \sim_r^2 f(q_1, q_2)$. Therefore, adding the patience rules for receiving arguments may preserve the weak 2-readiness equivalence.

In fact, we will assert, in the next section, that the patience rules for receiving arguments are necessary for the weak i-readiness equivalence with $i > 1$, but, they are not necessary for the weak 1-readiness equivalence.

4.3 Negative Premises Are Not Allowed

In fact, negative premises are hard to present in the congruence formats for any weak equivalences. This can be witnessed by a simple example shown in Figure 6: p_2 and q_2 satisfy almost all weak equivalences, including weak bisimulation. However, after adding rule $r = \dfrac{x_1 \xrightarrow{a} x_1', x_2 \stackrel{a}{\nrightarrow}}{f(x_1, x_2) \xrightarrow{d} g(x_1', x_2)}$ into language **B**, $f(p_1, p_2)$ and $f(q_1, q_2)$ are not even in weak trace equivalence.

Certainly, as stated in Section 2, we will employ the de Simone format as our starting point. And, no negative premises are allowed in the definition of de Simone format.

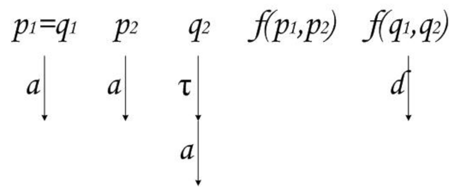

Figure 6: negative premises are not allowed in congruence formats for weak equivalences

5 Rule Formats for Weak Parametric Readiness Equivalence

This section will present formally the rule formats for the weak parametric readiness equivalences. In fact, as stated in the introduction, we will present three different rule formats: weak 1-readiness format for the weak 1-readiness equivalence, weak finite readiness format for the weak i-readiness equivalence with $1 < i < \omega$ and weak ω-readiness format for the weak ω-readiness equivalence.

5.1 Formal Definitions on the Rule Formats

Now, we put forward the weak 1-readiness format, which will be proved to be a congruence format for the weak 1-readiness equivalence.

Definition 5.1.1 A de Simone language $\mathcal{L}$ is in weak 1-readiness format if

1) patience rules are the only rules with τ-premises,

2) patience rules for active arguments of all operators are necessary,

3) rules with τ-conclusion are not permitted.

Following it, the weak finite readiness format is:

Definition 5.1.2 A de Simone language $\mathcal{L}$ is in weak finite readiness format if

1) patience rules are the only rules with τ-premises,

2) patience rules for active arguments and receiving arguments of all operators are all necessary,

3) rules with τ-conclusion are not permitted.

Then, the weak ω-readiness format for the weak ω-readiness equivalence is to be presented.

Definition 5.1.3 A de Simone language $\mathcal{L}$ is in weak ω-readiness format if

1) patience rules are the only rules with τ-premises, and

2) patience rules for active arguments and receiving arguments of all operators are all necessary.

It should be pointed out that the exclusion of rules with τ-conclusion and the allowance of dropping the patience rules for receiving arguments are not two separated restrictions. In fact, to obtain the effect of the allowance of dropping the patience rules for receiving arguments in the weak 1-readiness format, the exclusion of rules with τ-conclusion is a precondition. We will prove this conclusion in Lemma 5.3.5.

Below, we will study the relations between the above three rule formats. To fulfil this purpose, we need to define the 'tighter than' relation between rule formats.

Definition 5.1.4 Let A and B be two rule formats. A is tighter than B iff, for any language $\mathcal{L} = (\Sigma, \Psi)$ in format A, all transition rules in Ψ are also in format B. Moreover, A is strictly tighter than B iff A is tighter than B and there exists some languages $\mathcal{L} = (\Sigma, \Psi)$ in format B such that at least one of the transition rules in Ψ are not in format A.

Theorem 5.1.5 The weak 1-readiness format is strictly tighter than the weak finite readiness format and the weak finite readiness format is strictly tighter than the weak ω-readiness format.

Proof Comparing Definition 5.1.1 and Definition 5.1.2, patience rules for receiving arguments are not yet necessary for weak 1-readiness format. Therefore, for any languages $\mathcal{L}$ in weak 1-readiness format, its transition rules will also be in weak finite readiness format. Likewise, Definition 5.1.2 and Definition 5.1.3 are only different on the rules with τ-conclusion. Therefore, after refusing all rules with τ-conclusion, any languages $\mathcal{L}$ in weak finite readiness format will have its transition rules in the weak ω-readiness format.

The strictness between weak 1-readiness format and weak finite readiness format can be witnessed by the languages in Section 4.2. After introducing the patience rules for receiving arguments, it is a weak finite readiness language. However, patience rules for receiving arguments are not in weak 1-readiness format. The strictness between weak finite readiness format and weak ω-readiness format can be witnessed by introducing the hiding operator of CSP into any weak ω-readiness language. The obtained languages are still weak ω-readiness languages. However, one of transition rules of hiding operator is not in weak finite readiness format. $\square$

Observe that, the 'tighter than' relation is not the same as 'contained in' relation, which requires that, format A is contained in format B iff, any language $\mathcal{L} = (\Sigma, \Psi)$ in format A will also be in format B. It can be easily proved that, 'contained in' relation implies 'tighter than' relation, but not vice versa. Indeed, the weak 1-readiness format is tighter than the finite readiness format, but is not contained in the finite readiness format.

5.2 Ruloids And Ruloid Theorems On The Three Formats

Ruloids and the ruloid theorem originated from the works of Bloom [19, 22] for the GSOS format. In this subsection, we will introduce the ruloids and the ruloid theorem for the weak ω-readiness format. The ruloid theorems will be useful for proving the congruence theorems in the following subsections. The results obtained in this subsection will also hold for the other two formats.

For a language $\mathcal{L} = (\Sigma, \Psi)$ in the weak ω-readiness format, the ruloids $\mathscr{R}(C, \alpha)$, for a context C of n holes and an action α, are a set of expressions like the transition rules:

$$\frac{\{x_i \xrightarrow{\alpha_i} x'_i\}_{i \in I}}{C(x_1, ..., x_n) \xrightarrow{\alpha} D(y_1, ..., y_n)} \tag{4}$$

such that $y_i \equiv x'_i$ for $i \in I$ and $y_i \equiv x_i$ for $i \notin I$, where $I \subseteq \{1, 2, ..., n\}$. These expressions characterize all possible behaviors of the context C in the language. Besides, let $\mathscr{R}(C)$ denote the set of all ruloids of the context C of n holes, i.e., $\mathscr{R}(C) = \bigcup\{\mathscr{R}(C,\alpha)|\alpha \in Act \cup \{\tau\}\}$.

It should be noted that context D does not need to have exactly n holes. In fact, after leaving out the copying operation in the de Simone format (the weak ω-readiness format is a subformat of the de Simone format), the number of the holes of D should be less than or equivalent to n. But for convenience, in form (4), we still write it as $D(y_1, ..., y_n)$.

Furthermore, two properties should be imposed on $\mathscr{R}(C,\alpha)$, we call them soundness property and completeness property, by a bit of abusing the terminologies.

Definition 5.2.1 Let $\mathcal{L} = (\Sigma, \Psi)$ be a language in the weak ω-readiness format, and $C(x_1, ..., x_n)$ be any context of n holes in $\mathcal{L}$. A set $\mathscr{R}(C,\alpha)$ of ruloids of form (4) are ruloids of context C and action α, with $\alpha \in Act \cup \{\tau\}$, iff

1) Soundness. Let $r \in \mathscr{R}(C,\alpha)$ be a ruloid of form (4). If ζ is a closed Σ substitution such that $\zeta(x_i) \xrightarrow{\alpha_i} \zeta(x'_i)$ for all $i \in I$, then there must exist a context D such that $\zeta(C(x_1, ..., x_n)) \xrightarrow{\alpha} \zeta(D(y_1, ..., y_n))$.

2) Completeness. Let ζ be any closed Σ substitution. If $\zeta(C(x_1, ..., x_n)) \xrightarrow{\alpha}$, then there must exist a ruloid r of form (4) in ruloids $\mathscr{R}(C,\alpha)$, and $\zeta(x_i) \xrightarrow{\alpha_i}$ for all $i \in I$.

Below, we will present a strategy to retrieve the ruloids of context C and action α, and then prove that the obtained ruloids satisfy the above two properties, which form the ruloid theorem.

Strategy 5.2.2 Let $\mathcal{L} = (\Sigma, \Psi)$ be a language in the weak ω-readiness format. $C(x_1, ..., x_n)$ is any context of n holes in $\mathcal{L}$ and $\alpha \in Act \cup \{\tau\}$ is an action.

1) If $C \in V$, i.e., C is a variable, then $\mathscr{R}(C,\alpha) = \{\frac{x \xrightarrow{\alpha} x'}{x \xrightarrow{\alpha} x'}\}$;

2) If $C = f(x_1, ..., x_n)$ with $f \in \Sigma$ and $ar(f) = n$, then $\mathscr{R}(C,\alpha) = (f,\alpha)$, where (f,α) denotes the set of all rules in Ψ whose source is $f(x_1, ..., x_n)$ and output is α.

3) If C is any context. We can rewrite $C(x_1, ..., x_n)$ as $f(C_1(X_1), ..., C_m(X_m))$, where $f \in \Sigma$ and $ar(f) = m$. Note that $X_i \cap X_j = \emptyset$ with $1 \le i, j \le m$ and $i \ne j$. Without loss of generality, we may suppose that $X_i = x_{i1}x_{i2}...x_{im_i}$ for C_i is a context of m_i holes. Now, let r be any ruloid of form (4) in (f,α) and $\mathscr{R}(C_i,\alpha_i)$ be ruloids of context C_i and action α_i retrieved by induction on this strategy. Then, any ruloids in $\mathscr{R}(C,\alpha)$ can be obtained by the following steps:

i) pick out randomly from $\mathscr{R}(C_i,\alpha_i)$ a rule r_i, for all $i \in I$;

ii) substitute the variables x_j in r_i with x_{ij}, for all $1 \le j \le m_i$;

iii) substitute $x_i \xrightarrow{\alpha_i} x'_i$ in the premise of r with $ante(r_i)$, for all $i \in I$.

4) $\mathscr{R}(C,\alpha)$ is the set of all possible ruloids that can be retrieved from step 3). □

Theorem 5.2.3 Let $\mathcal{L} = (\Sigma, \Psi)$ be a language in the weak ω-readiness format, and $C(x_1, ..., x_n)$ be any context of n holes in $\mathcal{L}$. The set of ruloids $\mathscr{R}(C,\alpha)$ obtained from the Strategy 5.2.2 is the ruloids of context C and action α satisfies $\alpha \in Act \cup \{\tau\}$.

Proof Firstly, the obtained ruloids $\mathscr{R}(C,\alpha)$ of context C and action α are all in form (4), which can be easily retrieved from the construction procedure in the Strategy 5.2.2.

Secondly, the obtained ruloids $\mathscr{R}(C,\alpha)$ of context C and action α satisfy the soundness property. Let $r \in \mathscr{R}(C,\alpha)$ be a ruloid of form (4), where C is a context of n holes and $\alpha \in Act \cup \{\tau\}$ is an action. ζ is a closed Σ substitution such that $\zeta(x_i) \xrightarrow{\alpha_i} \zeta(x_i')$ for all $i \in I$.

i) if $C \in V$, then, without loss of generality, suppose $C = x$. The soundness property is trivial from $\mathscr{R}(C, \alpha) = \{\frac{x \xrightarrow{\alpha} x'}{x \xrightarrow{\alpha} x'}\}$;

ii) if $C = f(x_1, ..., x_n)$, then $\mathscr{R}(C, \alpha) = (f, \alpha)$. Therefore, the soundness property is guaranteed by the transition rules;

iii) if C is any context of n holes, then by Strategy 5.2.2, $C(x_1, ..., x_n)$ can be rewritten as $f(C_1(X_1), ..., C_m(X_m))$ for some operator $f \in \Sigma$ and $ar(f) = m$, and $ante(r)$ consist of $ante(r_1), ..., ante(r_m)$, where $r_i \in \mathscr{R}(C_i, \alpha_i)$ for all $1 \leq i \leq m$. By the assumption of the soundness property that $ante(r)$ is enabled under closed Σ substitution ζ. Therefore, by the induction hypothesis, $cons(r_1), ..., cons(r_m)$ are all enabled under ζ. This means that $\zeta(C_i(X_i)) \xrightarrow{\alpha_i} \zeta(D_i(Y_i))$ for all $1 \leq i \leq m$. In fact, $cons(r_1), ..., cons(r_m)$ constitute $ante(f)$. By the induction hypothesis on operator f, the transition rules in (f, α) guarantee the enableness of $C(x_1, ..., x_n) \xrightarrow{\alpha}$.

Last, the obtained ruloids $\mathscr{R}(C, \alpha)$ of context C and action α satisfy the completeness property, which can also be easily retrieved from the construction procedure of the Strategy 5.2.2. □

As we can see that, for a ruloid of form (4), its premises need not include all x_i for $1 \leq i \leq n$. However, we can add $x_i \xrightarrow{\epsilon} x_i'$, for $i \in \{1, ..., n\} \backslash I$, into the premises, as in the form (5). In this case, $\zeta(x_i) \xrightarrow{\epsilon} \zeta(x_i')$ denotes that subprocess $\zeta(x_i)$ executes no transition.

$$\frac{\{x_i \xrightarrow{\alpha_i} x_i'\}_{i \in I} \{x_i \xrightarrow{\epsilon} x_i'\}_{i \in \{1, ..., n\} - I}}{C(x_1, ..., x_n) \xrightarrow{\alpha} D(y_1, ..., y_n)} \tag{5}$$

The ϵ transition will not be added to the TSS. In fact, a TSS is a pair (Σ, Ψ), where Σ is a set of function symbols and Ψ is a set of transition rules assigned to the function symbols. Therefore, even no ruloids are in the TSS.

The introduction of ϵ transition is to substitute the ruloids of form (4) with the ruloids of form (5), since these two forms are equivalent when any closed Σ substitution ζ is applied. In fact, we want to express a viewpoint that, for any ruloid r, it should have two different but equivalent forms, i.e., form (4) and form (5).

For the equivalence between form (4) and form (5), we want to take an example to show it. Let $x_j \xrightarrow{\epsilon} x_j'$ be any ϵ-premise in some ruloid r. In fact, it denotes that, when ruloid r is applied in some Σ substitution ζ, subprocess $\zeta(x_j)$ is not fired at all. Also, if the form (4) of r is applied, the same results are retrieved.

The introduction of ϵ transitions and thus form (5) will make Lemma 5.3.1 and its proof prone to be comprehended. In Lemma 5.3.1, we will see that, in the weak ω-readiness languages, when process $C(p_1, ..., p_n)$ evolves into $C'(p_1, ..., p_n')$ by applying a ruloid and produce a transition (observable action or τ transition), each subprocess p_i will also evolve into p_i' and produce a transition (observable action, τ transition or ϵ transition).

Based on the ruloids and the ruloid theorem, we may restate several classes of rules, which have been defined previously, with the notion of ruloids. And, they will be more intuitive and prone to be used in the following.

The first class of rules which we concern is the patience rules. As their counterparts, the definition of patience ruloids is as follows.

Definition 5.2.4 A ruloid of the form $\dfrac{x_i \xrightarrow{\tau} x_i'}{C(x_1, ..., x_i, ..., x_n) \xrightarrow{\tau} C(x_1, ..., x_i', ..., x_n)}$ with $1 \leq i \leq n$ is called a patience ruloid of the ith argument of the context C.

In the following, a ruloid is called a plain ruloid if it is not a patience ruloid. Similar to the division in the patience rules, we also need to divide the patience ruloids into three classes, i.e., patience ruloids for active arguments, patience ruloids for receiving arguments and patience ruloids for other arguments.

In fact, Strategy 5.2.2 has already provided a canonical way to retrieve this division. Let $\mathcal{L}$ be a de Simone language and C be any context of n holes in it.

1) If only adding the patience rules for active arguments into the language, then, after using Strategy 5.2.2, the patience ruloids in $\mathscr{R}(C, \tau)$ are patience ruloids for active arguments.

2) If further adding the patience rules for receiving arguments into the language, then, after using Strategy 5.2.2, the patience ruloids in $\mathscr{R}(C, \tau)$ are patience ruloids for active arguments and receiving arguments. Therefore, getting rid of the patience ruloids for active arguments, we can obtain the patience ruloids for receiving arguments.

This division is obtained indirectly from Strategy 5.2.2 and patience rules, and thus it is hard to be used in the following. Here, we will propose another division which is directly based on the arguments of a context.

Definition 5.2.5 Let $\mathcal{L} = (\Sigma, \Psi)$ be a weak ω-readiness language, and C be any context of n holes. The ith argument of the context C is active if there exists a plain ruloid r of form (4) in $\mathscr{R}(C, \tau)$ such that x_i appears as left-hand side of a premise. The ith argument of the context C is receiving if it is not active and there exist another context D and a plain ruloid r of form (4) in $\mathscr{R}(D)$ such that $C(x_1', ..., x_n')$ appears as the target of r and x_i' appears as right-hand side of a premise.

Below, we will prove that these two divisions are indeed equivalent, i.e., a patience ruloid of some context C is a patience ruloid for active (resp. receiving, other) argument obtained from Strategy 5.2.2 and patience rules iff it is a patience ruloid for active (resp. receiving, other) argument defined by Definition 5.2.5.

Proposition 5.2.6 The division defined by Definition 5.2.5 is equivalent to the division obtained from Strategy 5.2.2 and patience rules.

Proof ($\Longleftarrow$) Let $\mathcal{L} = (\Sigma, \Psi)$ be a de Simone language, and C be any context of n holes.

If only adding the patience rules for active arguments into the language, we need to prove that each active argument of the context C defined by Definition 5.2.5 has a patience ruloid. We will prove by making an induction on the context C and Strategy 5.2.2.

1) If $C \in V$ or $C \in \Sigma$, then it can be easily obtained from Strategy 5.2.2 and Definition 2.8.

2) If C is any context, then it can be rewritten as $f(C_1(X_1), ..., C_m(X_m))$. Assume that contexts $C_1, ..., C_m$ satisfy that each active argument has a patience ruloid.

3) We need to prove that each active argument of C defined by Definition 5.2.5 has

a patience ruloid. Suppose that the ith argument of C is an active argument. Then, by Definition 5.2.5, there exists a plain ruloid r of form (4) in $\mathscr{R}(C, \tau)$ such that x_i appears as left-hand side of a premise. By Strategy 5.2.2, x_i must appear as left-hand side of a premise of some context. Without loss of generality, assume that x_i is the kth argument of the C_j. By the induction hypothesis, the kth argument of C_j is active and thus has a patience ruloid. Also by Strategy 5.2.2, the jth argument of functor f is active and thus has a patience ruloid. Therefore, we have that the ith argument of C has a patience ruloid by Strategy 5.2.2 and the above two patience ruloids for C_j and f, respectively.

If further adding the patience rules for receiving arguments into the language, we need to prove that each receiving argument of the context C defined by Definition 5.2.5 has a patience ruloid. Assume that the ith argument of context C is receiving. Then, by Definition 5.2.5, there exist another context D and a plain ruloid r of form (4) in $\mathscr{R}(D)$ such that $C(x'_1, ..., x'_n)$ appears as the target of r and x'_i appears as right-hand side of a premise. We will prove by making an induction on context C and Strategy 5.2.2.

1) If $C \in V$ or $C \in \Sigma$, then, by Definition 2.8, the ith argument of C is receiving. Therefore, it should have a patience rule by the hypothesis. By Strategy 5.2.2, each patience rule is also a patience ruloid.

2) If C is any context, then it can be rewritten as $f(C_1(X_1), ..., C_m(X_m))$. Assume that contexts $C_1, ..., C_m$ satisfy that each receiving argument has a patience ruloid.

3) We need to prove that the ith argument of C defined by Definition 5.2.5 has a patience ruloid. By Strategy 5.2.2, x_i must appear as right-hand side of a premise of some context. Without loss of generality, assume that x_i is the kth argument of the C_j. By the induction hypothesis, the kth argument of C_j is receiving or active and thus has a patience ruloid. Also by Strategy 5.2.2, the jth argument of functor f is receiving or active and thus has a patience ruloid. Therefore, we have that the ith argument of C has a patience ruloid by Strategy 5.2.2 and the above two patience ruloids for C_j and f, respectively.

($\Longrightarrow$) It is trivially true since, according to Strategy 5.2.2, each rule is also a ruloid. That is to say, we may first obtain the division on patience rules from the division on patience ruloids in Definition 5.2.5, and then using Strategy 5.2.2 to obtain the division from Strategy 5.2.2 and patience rules. □

The second class of rules is the rules with τ conclusion. Likewise, we may define the ruloids with τ conclusion as their counterparts.

Definition 5.2.7 A ruloid of the form $\dfrac{H}{C(x_1, ..., x_n) \xrightarrow{\tau} D(y_1, ..., y_n)}$ is called a ruloid with τ-conclusion, if it is not a patience ruloid and there exists at least one positive Σ literal in H.

Also, we want to show that the exclusion of rules with τ-conclusion is equivalent to the exclusion of ruloids with τ-conclusion.

Proposition 5.2.8 Let $\mathcal{L}$ be a weak ω-readiness language, and $C(x_1, ..., x_n)$ be any context of n holes. If, apart from patience rules, no rule with τ-conclusion is allowed, then no ruloid with τ-conclusion can be in $\mathscr{R}(C, \tau)$, and vice versa.

Proof This can be easily obtained from a fact that, by Strategy 5.2.2, the output of any ruloid is in fact the output of some rule. □

Table 1: The Usage of Lemmas in Section 5.3 in Proving the Congruence Theorems

format\ Lemma	5.3.1	5.3.2	5.3.3	5.3.4	5.3.5
1-readiness format	√	√		√	√
finite readiness format	√	√	√	√	
ω-readiness format	√	√	√		

5.3 Several Lemmas

In this subsection, several necessary lemmas are to be presented and the usage of them to prove the congruence theorems in the following three subsections has been listed in Table 1. The symbol √ in the table denotes that some lemma is to be used in the proof of the congruence theorem for the corresponding format. For example, the congruence theorem for the weak ω-readiness format needs the first three lemmas, i.e., Lemma 5.3.1, Lemma 5.3.2 and Lemma 5.3.3.

The following lemma states that, in the weak ω-readiness languages, any weak trace of a composite process may be decomposed into weak traces of its subprocesses. Besides, this lemma also holds in weak finite readiness languages and weak 1-readiness languages.

Lemma 5.3.1 Let $\mathcal{L} = (\Sigma, \Psi)$ be a weak ω-readiness language, and $C(x_1, ..., x_n)$ be any context of n holes. Suppose that ζ is any closed Σ substitution mapping x_i into p_i. If σ is a trace in $\mathcal{T}(C(p_1, ..., p_n), \omega)$, then, for all $1 \le i \le n$, there is a trace σ_i in $\mathcal{T}(p_i, \omega)$ such that, when $C(p_1, ..., p_n) \stackrel{\sigma}{\Longrightarrow} C'(p'_1, ..., p'_n)$, we have $p_i \stackrel{\sigma_i}{\Longrightarrow} p'_i$.

Proof Since $C(p_1, ..., p_n) \stackrel{\sigma}{\Longrightarrow} C'(p'_1, ..., p'_n)$, we have $C(p_1, ..., p_n) = C_0(p_{10}, ..., p_{n0}) \xrightarrow{\alpha_1} C_1(p_{11}, ..., p_{n1}) \xrightarrow{\alpha_2} ... \xrightarrow{\alpha_m} C_m(p_{1m}, ..., p_{nm}) = C'(p'_1, ..., p'_n)$, where $\forall 1 \le j \le m : \alpha_j \in Act \cup \{\tau\}$ and $\sigma' = \alpha_1 ... \alpha_m$ is equivalent to σ if all its τ transitions are omitted.

We will prove this lemma by making an induction on the length of σ'.

1) $|\sigma'| = 1$. Let $\sigma' = \alpha$. By the completeness property of the ruloids, there should be a ruloid of form (4) in $\mathscr{R}(C, \alpha)$, and $p_i \xrightarrow{\alpha_i}$ for all $i \in I$. As is shown before that, we have a ruloid of form (5) corresponding with form (4). Therefore, there exist $p_i \xrightarrow{\alpha_i} p'_i$ for all $i \in I$ and $p_i \xrightarrow{\epsilon} p'_i$ for all $i \in \{1, ..., n\} - I$, i.e., when $C(p_1, ..., p_n) \stackrel{\sigma}{\Longrightarrow} C'(p'_1, ..., p'_n)$, we have $p_i \stackrel{\alpha_i}{\Longrightarrow} p'_i$ for all $i \in I$ and $p_i \xrightarrow{\tau^*} p'_i$ for all $i \in \{1, ..., n\} - I$.

2) Assume that, when $|\sigma'| = m - 1$ with $m \ge 1$, if σ is a trace in $\mathcal{T}(C(p_1, ..., p_n), \omega)$ then, for all $1 \le i \le n$, there should be a trace σ_i in $\mathcal{T}(p_i, \omega)$ such that, when $C(p_1, ..., p_n) \stackrel{\sigma}{\Longrightarrow} C'(p'_1, ..., p'_n)$, we have $p_i \stackrel{\sigma_i}{\Longrightarrow} p'_i$.

3) For $|\sigma'| = m$, suppose that $C(p_1, ..., p_n) = C_0(p_{10}, ..., p_{n0}) \xrightarrow{\alpha_1} C_1(p_{11}, ..., p_{n1}) \xrightarrow{\alpha_2} ... \xrightarrow{\alpha_m} C_m(p_{1m}, ..., p_{nm}) = C'(p'_1, ..., p'_n)$. By the induction hypothesis, when $C(p_1, ..., p_n) = C_0(p_{10}, ..., p_{n0}) \xrightarrow{\alpha_1} C_1(p_{11}, ..., p_{n1}) \xrightarrow{\alpha_2} ... \xrightarrow{\alpha_{m-1}} C_{m-1}(p_{1(m-1)}, ..., p_{n(m-1)}) = C''(p''_1, ..., p''_n)$, we have $p_i \stackrel{\sigma''_i}{\Longrightarrow} p''_i$. Now, when $C_{m-1}(p_{1(m-1)}, ..., p_{n(m-1)}) = C''(p''_1, ..., p''_n) \xrightarrow{\alpha_m} C_m(p_{1m}, ..., p_{nm}) =$

$C'(p'_1, ..., p'_n)$, we have $p''_i \stackrel{\alpha'_m}{\Longrightarrow} p'_i$. Therefore, when $C(p_1, ..., p_n) \stackrel{\sigma}{\Longrightarrow} C'(p'_1, ..., p'_n)$, we have $p_i \stackrel{\sigma''_i \alpha'_m}{\Longrightarrow} p'_i$. □

The following lemma states that, in the weak ω-readiness languages, the weak trace equivalence will be preserved and the composite processes can reach the same contexts after same weak traces. The definition of weak trace equivalence is that: two processes p and q are weak trace equivalent, denoted as $p \sim_t q$, iff they have the same set of weak traces, i.e., $p \sim_t q$ iff $\mathcal{T}(p, \omega) = \mathcal{T}(q, \omega)$. This lemma also holds in weak finite readiness languages and weak 1-readiness languages.

Before that, we need one more definition on delay processes. Suppose that $\sigma \in \mathcal{T}(p, \omega)$ for some process p, then delay processes of $p \stackrel{\sigma}{\Longrightarrow}$ are those satisfying that 1) if $|\sigma| = 0$, then p itself is the delay process, and 2) if $|\sigma| \geq 1$, then let $\sigma = \sigma' a$ and delay processes are those processes p' such that $p \stackrel{\sigma}{\Longrightarrow}\stackrel{a}{\longrightarrow} p'$.

Lemma 5.3.2 Let $\mathcal{L} = (\Sigma, \Psi)$ be a weak ω-readiness language, and $C(x_1, ..., x_n)$ be any context of n holes. Suppose that ζ and ξ are any two closed Σ substitutions mapping x_i into p_i and q_i, respectively. If for all $1 \leq i \leq n$, $p_i \sim_t q_i$, then

1) for any trace $\sigma \in \mathcal{T}(C(p_1, ..., p_n), \omega)$ and some context C' such that $C(p_1, ..., p_n) \stackrel{\sigma}{\Longrightarrow} C'(p'_1, ..., p'_n)$, there exist $q'_1, ..., q'_n$ such that $C(q_1, ..., q_n) \stackrel{\sigma}{\Longrightarrow} C'(q'_1, ..., q'_n)$, and

2) if there exists a patience ruloid for the ith argument of context C' then q'_i can be any process such that $q_i \stackrel{\sigma_i}{\Longrightarrow} q'_i$, and if there does not exist a patience ruloid for the ith argument of context C' then q'_i can be any delay processes of $q_i \stackrel{\sigma_i}{\Longrightarrow}$, where σ_i is obtained by decomposing σ into the weak traces of subprocess p_i.

Proof Suppose that σ is a trace of $C(p_1, ..., p_n)$, and $C(p_1, ..., p_n) \stackrel{\sigma}{\Longrightarrow} C'(p'_1, ..., p'_n)$. By Lemma 5.3.1, when $C(p_1, ..., p_n) \stackrel{\sigma}{\Longrightarrow} C'(p'_1, ..., p'_n)$, we have $p_i \stackrel{\sigma_i}{\Longrightarrow} p'_i$ for all $1 \leq i \leq n$. Then, by $p_i \sim_t q_i$, we have $q_i \stackrel{\sigma_i}{\Longrightarrow}$ for all $1 \leq i \leq n$.

For $C(p_1, ..., p_n) \stackrel{\sigma}{\Longrightarrow} C'(p'_1, ..., p'_n)$, we have $C(p_1, ..., p_n) = C_0(p_{10}, ..., p_{n0}) \stackrel{\alpha_1}{\longrightarrow} C_1(p_{11}, ..., p_{n1}) \stackrel{\alpha_2}{\longrightarrow} ... \stackrel{\alpha_m}{\longrightarrow} C_m(p_{1m}, ..., p_{nm}) = C'(p'_1, ..., p'_n)$, where $\forall 1 \leq j \leq m : \alpha_j \in Act \cup \{\tau\}$ and $\sigma' = \alpha_1 ... \alpha_m$ is equivalent to σ if all its τ transitions are omitted.

Suppose that the sequence of plain ruloids applied in the above procedure is $r_1 r_2 ... r_k$. It is enough to show that $C(q_1, ..., q_n)$ can also apply ruloids $r_1 r_2 ... r_k$ in the same order, and $C(q_1, ..., q_n) \stackrel{\sigma}{\Longrightarrow} C'(q'_1, ..., q'_n)$ for some $q'_1, ..., q'_n$. Furthermore, if there exists a patience ruloid for the ith argument of context C' then q'_i can be any process such that $q_i \stackrel{\sigma_i}{\Longrightarrow} q'_i$, and if there does not exist a patience ruloid for the ith argument of context C' then q'_i can be any delay processes of $q_i \stackrel{\sigma_i}{\Longrightarrow}$.

We will prove it by making an induction on k.

1) $k = 0$. Then, $C = C'$ and only patience ruloids are applied when $C(p_1, ..., p_n) \stackrel{\sigma}{\Longrightarrow} C'(p'_1, ..., p'_n)$ and thus $\sigma = \tau^*$. By Lemma 5.3.1, $\sigma_i = \tau^*$ for all $1 \leq i \leq n$. Therefore, there must exist $q'_i, ..., q'_n$ such that $C(q_1, ..., q_n) \stackrel{\tau^*}{\Longrightarrow} C'(q'_1, ..., q'_n)$ since an extreme possibility is that $q_i \equiv q'_i$ for all $1 \leq i \leq n$. Now, if there exists a patience ruloid for the ith argument of context C' then q'_i can be any process such that $q_i \stackrel{\tau^*}{\longrightarrow} q'_i$ by the soundness property of ruloids and the definition of patience ruloids. On the other hand,

if there does not exist a patience ruloid for the ith argument of context C' then q'_i can be q_i.

2) Assume that, when $k = m - 1$ with $m \geq 1$, the above statement holds.

3) For $k = m$, suppose that, $C(p_1, ..., p_n) \overset{\sigma}{\Longrightarrow} C''(p''_1, ..., p''_n)$ and $C''(p''_1, ..., p''_n) \overset{\delta}{\Longrightarrow} C'(p'_1, ..., p'_n)$, where the first $k-1$ plain ruloids of $r_1 r_2 ... r_k$ are applied when $C(p_1, ..., p_n) \overset{\sigma}{\Longrightarrow} C''(p''_1, ..., p''_n)$ and the kth plain ruloid are applied when $C''(p''_1, ..., p''_n) \overset{\delta}{\Longrightarrow} C'(p'_1, ..., p'_n)$.

By Lemma 5.3.1, there exist σ_i, δ_i for all $1 \leq i \leq n$ such that $p_i \overset{\sigma_i}{\Longrightarrow} p''_i$ and $p''_i \overset{\delta_i}{\Longrightarrow} p'_i$. By $p_i \sim_t q_i$, we have $q_i \overset{\sigma_i\delta_i}{\Longrightarrow}$.

Then, by the induction hypothesis, $C(q_1, ..., q_n)$ can also apply the first $k-1$ ruloids and reaches $C''(q''_1, ..., q''_n)$. Moreover, if there exists a patience ruloid for the ith argument of context C'' then q''_i can be any process such that $q_i \overset{\sigma_i}{\Longrightarrow} q''_i$, and if there does not exist a patience ruloid for the ith argument of context C'' then q''_i can be any delay process of $q_i \overset{\sigma_i}{\Longrightarrow}$.

Furthermore, for all $1 \leq i \leq n$, let q''_i be any delay process of $q_i \overset{\sigma_i}{\Longrightarrow}$ such that $q_i \overset{\sigma_i}{\Longrightarrow} q''_i \overset{\delta_i}{\Longrightarrow}$. There always exists a such q''_i because of $q_i \overset{\sigma_i\delta_i}{\Longrightarrow}$.

Suppose that the kth ruloid r_k is in form (4). Then, by the definition of the weak ω-readiness format, all arguments in I have corresponding patience ruloids since they are all active arguments of C'' by Definition 5.2.5. Therefore, by the soundness property of the ruloids, we may apply the patience ruloids for the arguments in I and obtain $C''(q''_1, ..., q''_n) \overset{\tau^*}{\Longrightarrow} C''(q'''_1, ..., q'''_n)$, such that $q'''_i \equiv q''_i$ if $i \notin I$ and $q'''_i \overset{\delta_i}{\longrightarrow}$ if $i \in I$. Then, also by the soundness property of the ruloids, ruloid r_k will be applied and $C''(q'''_1, ..., q'''_n) \overset{\delta}{\longrightarrow} C'(q''''_1, ..., q''''_n)$, where $q''''_i \equiv q''_i$ if $i \notin I$ and q''''_i is any process satisfying $q'''_i \overset{\delta_i}{\longrightarrow} q''''_i$ if $i \in I$.

Now, we can see that q''''_i is indeed a delay process of $q_i \overset{\sigma_i\delta_i}{\Longrightarrow}$.

Finally, if there exists a patience ruloid for the ith argument of context C' then q''''_i may evolve into any process q'_i such that $q''''_i \overset{\tau^*}{\Longrightarrow} q'_i$ and thus q'_i may be any process such that $q_i \overset{\sigma_i\delta_i}{\Longrightarrow} q'_i$. On the other hand, if there does not exist a patience ruloid for the ith argument of context C' then let q'_i be q''''_i, and thus q'_i is any delay process of $q_i \overset{\sigma_i\delta_i}{\Longrightarrow}$. $\square$

As a strengthened results of the above lemma, Lemma 5.3.3 below will show that, in weak finite readiness languages and weak ω-readiness languages, if the ith argument is neither an active argument nor a receiving argument, i.e., is an other argument, then q'_i can be q_i or any process such that $q_i \overset{\sigma_i}{\Longrightarrow} q'_i$. Though we only prove this lemma in weak ω-readiness languages, it also holds in weak finite readiness languages.

Lemma 5.3.3 Let $\mathcal{L} = (\Sigma, \Psi)$ be a weak ω-readiness language, and $C(x_1, ..., x_n)$ be any context of n holes. Suppose that ζ and ξ are any two closed Σ substitution mapping x_i into p_i and q_i, respectively. For all $1 \leq i \leq n$, $p_i \sim_t q_i$, and thus for any trace $\sigma \in \mathcal{T}(C(p_1, ..., p_n), \omega)$ and some context C' with $C(p_1, ..., p_n) \overset{\sigma}{\Longrightarrow} C'(p'_1, ..., p'_n)$, there exist $q'_1, ..., q'_n$ such that $C(q_1, ..., q_n) \overset{\sigma}{\Longrightarrow} C'(q'_1, ..., q'_n)$. Now, if the ith argument of C'

is an other argument, then q_i' can be q_i or any process such that $q_i \overset{\sigma_i}{\Longrightarrow} q_i'$.

Note: Here, the word 'other' may incur misunderstandings. In fact, it denotes that the division of the ith argument is neither an active argument nor a receiving argument.

Proof Like the proof of Lemma 5.3.2, suppose that the sequence of plain ruloids applied in the procedure $C(p_1, ..., p_n) \overset{\sigma}{\Longrightarrow} C'(p_1', ..., p_n')$ is $r_1 r_2 ... r_k$.

We will prove it by making an induction on k.

1) $k = 0$. Then, $C = C'$ and only patience ruloids are applied when $C(p_1, ..., p_n) \overset{\sigma}{\Longrightarrow} C'(p_1', ..., p_n')$ and thus $\sigma = \tau^*$. In this case, we can let q_i' be q_i if the ith argument of C' is an other argument.

2) Assume that, when $k = m - 1$ with $m \geq 1$, the lemma holds.

3) For $k = m$, suppose that, $C(p_1, ..., p_n) \overset{\sigma}{\Longrightarrow} C''(p_1'', ..., p_n'')$ and $C''(p_1'', ..., p_n'') \overset{\delta}{\Longrightarrow} C'(p_1', ..., p_n')$, where the first $k-1$ plain ruloids of $r_1 r_2 ... r_k$ is applied when $C(p_1, ..., p_n) \overset{\sigma}{\Longrightarrow} C''(p_1'', ..., p_n'')$ and the kth plain ruloid are applied when $C''(p_1'', ..., p_n'') \overset{\delta}{\Longrightarrow} C'(p_1', ..., p_n')$.

However, observe that the ith argument of C'' cannot be in set I of ruloid r_k. Or else, the ith argument of C' will be at least a receiving argument by Definition 5.2.5. We separate it into two cases:

i) If the ith argument of C'' is an active argument or a receiving argument, then it has a patience ruloid since the language $\mathcal{L}$ is a weak ω-readiness language. Therefore, by Lemma 5.3.2, q_i' can be any process such that $q_i \overset{\sigma_i \delta_i}{\Longrightarrow} q_i'$ with $\delta_i = \tau^*$.

ii) If the ith argument of C'' is an other argument, then, by the induction hypothesis, q_i'' can be q_i or any process such that $q_i \overset{\sigma_i}{\Longrightarrow} q_i''$. Now, since $\mathcal{L}$ is a weak ω-readiness language, no patience ruloid for the ith argument of C'' and the ith argument of C'. Therefore, q_i' is just q_i'', and thus q_i' can be q_i or any process such that $q_i \overset{\sigma_i}{\Longrightarrow} q_i''$. Then, by $\delta_i = \tau^*$, q_i' can be q_i or any process such that $q_i \overset{\sigma_i \delta_i}{\Longrightarrow} q_i'$. $\square$

Note that, the above lemma does not hold in weak 1-readiness languages since patience ruloids for receiving arguments are needed when proving it. Therefore, it will not be used when proving the congruence theorem for the wean 1-readiness format.

The following lemma shows that, in a weak finite readiness language, if a process executes an action sequence (weak trace) with length k, then, at the same time, all the lengths of the action sequences executed by its subprocesses may not exceed k. This lemma also holds in weak 1-readiness languages.

Lemma 5.3.4 Let $\mathcal{L}$ be a weak finite readiness language. $C(x_1, ..., x_n)$ is any context of n holes in $\mathcal{L}$. Suppose that ζ is any closed Σ substitution mapping x_i into p_i. If $C(p_1, ..., p_n)$ is a process and σ is a weak trace of $C(p_1, ..., p_n)$, then each p_i will execute a weak trace σ_i at the same time, for $1 \leq i \leq n$. We can conclude that $\forall 1 \leq i \leq n$: $|\sigma_i| \leq k$ when $|\sigma| = k$.

Proof By Lemma 5.3.1, if $C(p_1, ..., p_n)$ is a process and σ is a weak trace of $C(p_1, ..., p_n)$, then each p_i will execute a weak trace σ_i at the same time. We also need to prove that $|\sigma_i| \leq k$ when $|\sigma| = k$.

We will prove it by making an induction on $|\sigma| = k$.

1) $k = 0$. Then, $C(p_1, ..., p_n) \overset{\tau^*}{\Longrightarrow} C'(p_1', ..., p_n')$. By the definition of the weak ω-readiness format, the ruloids applied in this procedure can only be patience ruloids or

ruloids with τ conclusion. By the hypothesis, no rules with τ conclusion are present in $\mathcal{L}$, and thus, by Proposition 5.2.8, no ruloids with τ conclusion are present in $\mathscr{R}(C,\tau)$. Therefore, when $C(p_1,...,p_n) \stackrel{\tau^*}{\Longrightarrow} C'(p'_1,...,p'_n)$, only patience ruloids are applied. However, from the definition of patience ruloids and its corresponding ruloids in form (5), $p_i \xrightarrow{\tau} p'_i$ or $p_i \xrightarrow{\epsilon} p'_i$ for all $1 \leq i \leq n$. And $|\tau^*| = |\epsilon| = 0$.

2) Assume that, when $k = m-1$ with $m \geq 1$, we have $|\sigma_i| \leq k$ when $|\sigma| = k$.

3) For $k = m$, let $\sigma = \sigma'\alpha$. Then, we have $C(p_1,...,p_n) \stackrel{\sigma'}{\Longrightarrow} C''(p''_1,...,p''_n) \stackrel{\alpha}{\Longrightarrow} C'(p'_1,...,p'_n)$. Extending it, we obtain that $C(p_1,...,p_n) \stackrel{\sigma'}{\Longrightarrow} C''(p''_1,...,p''_n) \xrightarrow{\tau^*} C^1(p^1_1,...,p^1_n) \xrightarrow{\alpha} C^2(p^2_1,...,p^2_n) \xrightarrow{\tau^*} C'(p'_1,...,p'_n)$.

By Lemma 5.3.1, for all $1 \leq i \leq n$, there exist $p_i \stackrel{\sigma'_i}{\Longrightarrow} p''_i \stackrel{\sigma^1_i}{\Longrightarrow} p^1_i \stackrel{\alpha_i}{\Longrightarrow} p^2_i \stackrel{\sigma^2_i}{\Longrightarrow} p'_i$.

It is trivial that $|\sigma'| = m-1$. Therefore, by the induction hypothesis, we have $|\sigma'_i| \leq m-1$. Also, when $C''(p''_1,...,p''_n) \xrightarrow{\tau^*} C^1(p^1_1,...,p^1_n)$ and $C^2(p^2_1,...,p^2_n) \xrightarrow{\tau^*} C'(p'_1,...,p'_n)$, $|\tau^*| = 0$. Therefore, by the induction base, we have $|\sigma^1_i| = |\sigma^2_i| = 0$. Furthermore, $|\alpha_i| \leq 1$ can be obtained by the ruloids of form (5).

In all, $|\sigma_i| \leq k$ when $|\sigma| = k$. □

The following lemma shows that, in weak 1-readiness languages, processes $C(p_1,...,p_i,...,p_n)$ and $C(p_1,...,p'_i,...,p_n)$ have the same sets of next observable actions if $p_i \xrightarrow{\tau^*} p'_i$ and the ith argument is not an active argument.

Lemma 5.3.5 Let $\mathcal{L} = (\Sigma,\Psi)$ be a weak 1-readiness language, and $C(x_1,...,x_n)$ be any context of n holes. Suppose that ζ is any closed Σ substitution mapping x_i into p_i. If the ith argument is not an active argument of $C(x_1,...,x_n)$ and $p_i \xrightarrow{\tau^*} p'_i$, then $\mathcal{T}(C(p_1,...,p_i,...,p_n),1) = \mathcal{T}(C(p_1,...,p'_i,...,p_n),1)$.

Proof Without loss of generality, suppose that $p = C(p_1,...p_i,...,p_n)$ and $q = C(p_1,...,p'_i,...,p_n)$, where C is any context of n holes in the language $\mathcal{L}$. Let $A_1 = \{a \in Act | p \stackrel{a}{\Longrightarrow}\}$ and $A_2 = \{a \in Act | p \stackrel{a}{\Longrightarrow}\}$. We need to prove $A_1 = A_2$. Consider the next ruloid which will be applied.

1) If the next ruloid is a patience ruloid, then it should be a patience ruloid for active argument, since $\mathcal{L}$ is a weak 1-readiness language. However, applying the patience ruloid will not produce observable actions for $C(p_1,...p_i,...,p_n)$ and $C(p_1,...,p'_i,...,p_n)$. Because the ith argument is not an active argument, $C(p_1,...p_i,...,p_j,...,p_n) \xrightarrow{\tau} C(p_1,...p_i,...,p'_j,...,p_n)$ and $C(p_1,...p'_i,...,p_j,...,p_n) \xrightarrow{\tau} C(p_1,...p'_i,...,p'_j,...,p_n)$ when the jth argument of context C is an active argument and $p_j \xrightarrow{\tau} p'_j$. Now, it is enough to consider the set of next observable actions of $C(p_1,...p_i,...,p'_j,...,p_n)$ and $C(p_1,...p'_i,...,p'_j,...,p_n)$.

2) If the next ruloid is a plain ruloid, then it should not be a ruloid with τ conclusion, since $\mathcal{L}$ is a weak 1-readiness language. Suppose that the applied ruloid r is in form (4), then the ith argument is not in I since it is not an active argument. Therefore, by the soundness property of the ruloids, the p_i will not be fired when applying the ruloid r. Furthermore, since p and q are only different in p_i and p'_i, we have $A_1 = A_2$. □

5.4 Weak 1-Readiness Format for Weak 1-Readiness Equivalence

Now, we will prove the congruence theorem for the weak 1-readiness format.

Theorem 5.4.1 The weak 1-readiness format is a congruence format for the weak 1-readiness equivalence.

Proof It is enough to prove that if $\forall 1 \leq j \leq n : p_j \sim_r^1 q_j$ then $C(p_1, ..., p_n) \sim_r^1 C(q_1, ..., q_n)$, where C is any context of n holes in a weak 1-readiness language $\mathcal{L}$. By the symmetry, we only need to prove that if $\forall 1 \leq j \leq n : p_j \sim_r^1 q_j$, then $\mathcal{R}(C(p_1, ..., p_n), 1) \subseteq \mathcal{R}(C(q_1, ..., q_n), 1)$.

Suppose that (σ, A) is any weak 1-readiness pair in $\mathcal{R}(C(p_1, ..., p_n), 1)$, we need to show that $(\sigma, A) \in \mathcal{R}(C(q_1, ..., q_n), 1)$. By $(\sigma, A) \in \mathcal{R}(C(p_1, ..., p_n), 1)$ and the definition of weak 1-readiness pair, there exists $C'(p'_1, ..., p'_n)$ such that $C(p_1, ..., p_n) \stackrel{\sigma}{\Longrightarrow} C'(p'_1, ..., p'_n) \wedge \mathcal{T}(C'(p'_1, ..., p'_n), 1) = A$.

By Lemma 5.3.1, when $C(p_1, ..., p_n) \stackrel{\sigma}{\Longrightarrow} C'(p'_1, ..., p'_n)$, there exists $p_j \stackrel{\sigma_j}{\Longrightarrow} p'_j$ for any subprocess p_j with $1 \leq j \leq n$. Similarly, for all $a \in A$, when $C'(p'_1, ..., p'_n) \stackrel{a}{\Longrightarrow}$, we have $p'_j \stackrel{\delta_j}{\Longrightarrow}$ for all subprocess p_j with $1 \leq j \leq n$. Let A'_j be the set of all δ_j. Note that, for some $a \in A$, there may exist several δ_j corresponding with it. And we should add all of them into the set A'_j.

Then, by Lemma 5.3.4, the exclusion of the rules with τ-conclusion will make the length of δ_j not exceed 1, i.e., $\forall \delta_j \in A'_j : |\delta_j| \leq 1$. Therefore, for all $1 \leq j \leq n$, we have $(\sigma_j, A_j) \in \mathcal{R}(p_j, 1)$ such that $A'_j \subseteq A_j$ and $A_j = \mathcal{T}(p'_j, 1)$.

Now, by $p_j \sim_r^1 q_j$, we have $(\sigma_j, A_j) \in \mathcal{R}(q_j, 1)$. Without loss of generality, suppose that $q_j \stackrel{\sigma_j}{\Longrightarrow} q'_j$ and $\mathcal{T}(q'_j, 1) = A_j$.

By Lemma 5.3.2, σ is also a trace of $C(q_1, ..., q_n)$ and $C(q_1, ..., q_n) \stackrel{\sigma}{\Longrightarrow} C'(q''_1, ..., q''_n)$. Observe that, it is possible that q''_j is not equivalent to q'_j. The reason is that, from Lemma 5.3.2, we can only obtain that, there exist $q''_1, ..., q''_n$ such that $C(q_1, ..., q_n) \stackrel{\sigma}{\Longrightarrow} C'(q''_1, ..., q''_n)$, but not the very $q'_1, ..., q'_n$ which are obtained from $(\sigma_j, A_j) \in \mathcal{R}(q_j, 1)$.

By Lemma 5.3.2, if the jth argument of C' is an active argument, then q''_j can be any process such that $q_j \stackrel{\sigma_j}{\Longrightarrow} q''_j$ and thus we may let q''_j be q'_j safely because of $q_j \stackrel{\sigma_j}{\Longrightarrow} q'_j$. On the other hand, if the jth argument of C' is not an active argument, then no patience ruloids are present in the language. Therefore, q''_j can be any delay process of $q_i \stackrel{\sigma_j}{\Longrightarrow}$. Note that, for q'_j, there must exist some delay process q''_j such that $q_j \stackrel{\sigma_j}{\Longrightarrow} q''_j \stackrel{\tau^*}{\longrightarrow} q'_j$.

Now, by Lemma 5.3.5 and $q''_j \stackrel{\tau^*}{\longrightarrow} q'_j$, we assert that $\mathcal{T}(C'(q''_1, ..., q''_n), 1) = \mathcal{T}(C'(q'_1, ..., q'_n), 1)$. Note that, there may exist several arguments of C' such that they are not active arguments. However, we can finally obtain $\mathcal{T}(C'(q''_1, ..., q''_n), 1) = \mathcal{T}(C'(q'_1, ..., q'_n), 1)$ by applying Lemma 5.3.5 for several times.

Moreover, by $\mathcal{T}(q'_j, 1) = \mathcal{T}(p'_j, 1) = A_j$, we have $\mathcal{T}(C'(q'_1, ..., q'_n), 1) = \mathcal{T}(C'(p'_1, ..., p'_n), 1)$.

Finally, we obtain that $\mathcal{T}(C'(q''_1, ..., q''_n), 1) = \mathcal{T}(C'(p'_1, ..., p'_n), 1) = A$, and thus $(\sigma, A) \in \mathcal{R}(C(q_1, ..., q_n), 1)$. $\square$

5.5 Weak Finite Readiness Format for Weak i-Readiness Equivalence

The following is the congruence theorem for the weak finite readiness format.

Theorem 5.5.1 The weak finite readiness format is a congruence format for the weak i-readiness equivalence with $1 < i < \omega$.

Proof It is enough to prove that if $\forall 1 \leq j \leq n : p_j \sim_r^i q_j$ then $C(p_1, ..., p_n) \sim_r^i C(q_1, ..., q_n)$, where C is any context of n holes in a weak finite readiness language $\mathcal{L}$. By the symmetry, we only need to prove that if $\forall 1 \leq j \leq n : p_j \sim_r^i q_j$, then $\mathcal{R}(C(p_1, ..., p_n), i) \subseteq \mathcal{R}(C(q_1, ..., q_n), i)$.

Suppose that (σ, Φ) is any weak i-readiness pair in $\mathcal{R}(C(p_1, ..., p_n), i)$, we need to show that $(\sigma, \Phi) \in \mathcal{R}(C(q_1, ..., q_n), i)$. By $(\sigma, \Phi) \in \mathcal{R}(C(p_1, ..., p_n), i)$ and the definition of weak i-readiness pair, there exists $C'(p'_1, ..., p'_n)$ such that $C(p_1, ..., p_n) \stackrel{\sigma}{\Longrightarrow} C'(p'_1, ..., p'_n) \wedge \mathcal{T}(C'(p'_1, ..., p'_n), i) = \Phi$.

By Lemma 5.3.1, when $C(p_1, ..., p_n) \stackrel{\sigma}{\Longrightarrow} C'(p'_1, ..., p'_n)$, there exists $p_j \stackrel{\sigma_j}{\Longrightarrow} p'_j$ for all subprocess p_j with $1 \leq j \leq n$. Similarly, for all $\delta \in \Phi$, when $C'(p'_1, ..., p'_n) \stackrel{\delta}{\Longrightarrow}$, we have $p'_j \stackrel{\delta_j}{\Longrightarrow}$ for all subprocess p_j with $1 \leq j \leq n$. Let Φ'_j be the set of all δ_j. Note that, for some $\delta \in \Phi$, there may exist several δ_j corresponding with it. And we should add all of them into the set Φ'_j.

By Lemma 5.3.4, the exclusion of the rules with τ-conclusion will make the length of δ_j not exceed i, i.e., $\forall \delta_j \in \Phi'_j : |\delta_j| \leq i$. Therefore, for all $1 \leq j \leq n$, we have $(\sigma_j, \Phi_j) \in \mathcal{R}(p_j, i)$ such that $\Phi'_j \subseteq \Phi_j$ and $\Phi_j = \mathcal{T}(p'_j, i)$.

Now, by $p_j \sim_r^i q_j$, we have $(\sigma_j, \Phi_j) \in \mathcal{R}(q_j, i)$. Without loss of generality, suppose that $q_j \stackrel{\sigma_j}{\Longrightarrow} q'_j$ and $\mathcal{T}(q'_j, i) = \Phi_j$.

By Lemma 5.3.2, σ is also a trace of $C(q_1, ..., q_n)$ and $C(q_1, ..., q_n) \stackrel{\sigma}{\Longrightarrow} C'(q''_1, ..., q''_n)$. Moreover,

1) if the jth argument of C' is a receiving argument or an active argument, then it has a patience ruloid since the language is a weak finite readiness language. Therefore, by Lemma 5.3.2, we can let q''_j be q'_j since q''_j can be any process such that $q_j \stackrel{\sigma_i}{\Longrightarrow} q''_j$, and

2) if the jth argument of C' is an other argument, then, by Lemma 5.3.3, we can let q''_j be q_j or any process such that $q_j \stackrel{\sigma_j}{\Longrightarrow} q''_j$. We want to separate it into two cases:

i) if q''_j is any process such that $q_j \stackrel{\sigma_j}{\Longrightarrow} q''_j$, then we can also let q''_j be q'_j.

ii) if q''_j is q_j, then q'_j and q''_j are both delay processes since the jth argument of C' is an other argument and thus no patience ruloid for it. Therefore, we can obtain that $q''_j \equiv q'_j \equiv q_j$.

In all, we can always let q''_j be q'_j, i.e., $C(q_1, ..., q_n) \stackrel{\sigma}{\Longrightarrow} C'(q'_1, ..., q'_n)$.

Moreover, by $\mathcal{T}(q'_j, i) = \mathcal{T}(p'_j, i) = \Phi_j$, we have $\mathcal{T}(C'(q'_1, ..., q'_n), i) = \mathcal{T}(C'(p'_1, ..., p'_n), i)$, and thus $(\sigma, \Phi) \in \mathcal{R}(C(q_1, ..., q_n), i)$. $\square$

5.6 Weak ω-Readiness Format for Weak ω-Readiness Equivalence

The congruence theorem for the weak ω-readiness format is as follows.

Theorem 5.6.1 The weak ω-readiness format is a congruence format for the weak ω-readiness equivalence.

Proof It is enough to prove that if $\forall 1 \leq j \leq n : p_j \sim_r^\omega q_j$ then $C(p_1, ..., p_n) \sim_r^\omega C(q_1, ..., q_n)$, where C is any context of n holes in a weak finite readiness language $\mathcal{L}$. By the symmetry, we only need to prove that if $\forall 1 \leq j \leq n : p_j \sim_r^\omega q_j$, then $\mathcal{R}(C(p_1, ..., p_n), \omega) \subseteq \mathcal{R}(C(q_1, ..., q_n), \omega)$.

Suppose that (σ, Φ) is any weak ω-readiness pair in $\mathcal{R}(C(p_1, ..., p_n), \omega)$, we need to show that $(\sigma, \Phi) \in \mathcal{R}(C(q_1, ..., q_n), \omega)$. By $(\sigma, \Phi) \in \mathcal{R}(C(p_1, ..., p_n), \omega)$ and the definition of weak ω-readiness pair, there exists $C'(p'_1, ..., p'_n)$ such that $C(p_1, ..., p_n) \stackrel{\sigma}{\Longrightarrow} C'(p'_1, ..., p'_n) \wedge \mathcal{T}(C'(p'_1, ..., p'_n), \omega) = \Phi$.

By Lemma 5.3.1, when $C(p_1, ..., p_n) \stackrel{\sigma}{\Longrightarrow} C'(p'_1, ..., p'_n)$, there exists $p_j \stackrel{\sigma_j}{\Longrightarrow} p'_j$ for all subprocess p_j with $1 \leq j \leq n$. Similarly, for all $\delta \in \Phi$, when $C'(p'_1, ..., p'_n) \stackrel{\delta}{\Longrightarrow}$, we have $p'_j \stackrel{\delta_j}{\Longrightarrow}$ for all subprocess p_j with $1 \leq j \leq n$. Let Φ'_j be the set of all δ_j. Note that, for some $\delta \in \Phi$, there may exist several δ_j corresponding with it. And we should add all of them into the set Φ'_j.

Therefore, for all $1 \leq j \leq n$, we have $(\sigma_j, \Phi_j) \in \mathcal{R}(p_j, \omega)$ such that $\Phi'_j \subseteq \Phi_j$ and $\Phi_j = \mathcal{T}(p'_j, \omega)$.

Now, by $p_j \sim_r^\omega q_j$, we have $(\sigma_j, \Phi_j) \in \mathcal{R}(q_j, \omega)$. Without loss of generality, suppose that $q_j \stackrel{\sigma_j}{\Longrightarrow} q'_j$ and $\mathcal{T}(q'_j, \omega) = \Phi_j$.

By Lemma 5.3.2, σ is also a trace of $C(q_1, ..., q_n)$ and $C(q_1, ..., q_n) \stackrel{\sigma}{\Longrightarrow} C'(q''_1, ..., q''_n)$. Moreover,

1) if the jth argument of C' is a receiving argument or an active argument, then it has a patience ruloid since the language is a weak ω-readiness language. Therefore, by Lemma 5.3.2, we can let q''_j be q'_j since q''_j can be any process such that $q_j \stackrel{\sigma_i}{\Longrightarrow} q''_j$, and

2) if the jth argument of C' is an other argument, then, by Lemma 5.3.3, we can let q''_j be q_j or any process such that $q_j \stackrel{\sigma_j}{\Longrightarrow} q''_j$. We want to separate it into two cases:

i) if q''_j is any process such that $q_j \stackrel{\sigma_j}{\Longrightarrow} q''_j$, then we can also let q''_j be q'_j.

ii) if q''_j is q_j, then q'_j and q''_j are both delay processes since the jth argument of C' is an other argument and thus no patience ruloid for it. Therefore, we can obtain that $q''_j \equiv q'_j \equiv q_j$.

In all, we can always let q''_j be q'_j, i.e., $C(q_1, ..., q_n) \stackrel{\sigma}{\Longrightarrow} C'(q'_1, ..., q'_n)$.

Moreover, by $\mathcal{T}(q'_j, \omega) = \mathcal{T}(p'_j, \omega) = \Phi_j$, we have $\mathcal{T}(C'(q'_1, ..., q'_n), \omega) = \mathcal{T}(C'(p'_1, ..., p'_n), \omega)$, and thus $(\sigma, \Phi) \in \mathcal{R}(C(q_1, ..., q_n), \omega)$. $\square$

6 Conclusions

In the paper, we first introduce a series of behavioral equivalences, named weak parametric readiness equivalences, which take the weak readiness equivalence and the weak

Table 2: weak parametric readiness equivalences and their corresponding rule formats

equivalence	rule format	concise characterizations
weak ω-readiness	weak ω-readiness	de Simone + patience rules for active arguments and receiving arguments
...	...	...
weak i-readiness	weak finite readiness	rules with τ-conclusion are not allowed
...	...	...
weak 2-readiness	weak finite readiness	rules with τ-conclusion are not allowed
weak 1-readiness	weak 1-readiness	patience rules for receiving arguments are not necessary

possible future equivalence as their special cases. Then, based on the idea of Structural Operational Semantics, the rule formats are proposed to meet these behavioral equivalences. By the formal proofs, we have shown that these rule formats are all congruence formats of their corresponding behavioral equivalences.

The corresponding relations between the weak parametric readiness equivalences and their corresponding rule formats have been collected in the Table 2. This table reflects the gradual changes among the weak parametric readiness equivalences in that, when the weak parametric readiness equivalences become coarser, their corresponding rule formats turn tighter. Firstly, the weak ω-readiness format for the weak ω-readiness equivalence is denoted concisely as 'de Simone + patience rules for active arguments and receiving arguments'. Then, the weak finite readiness format for the weak i-readiness equivalence with $1 < i < \omega$ should exclude the rules with τ-conclusion. At the same time, the weak i-readiness is strictly coarser than the weak ω-readiness equivalence. Finally, the weak 1-readiness format for the weak 1-readiness equivalence may further exclude the patience rules for receiving arguments, when the weak 1-readiness equivalence is strictly coarser than the weak i-readiness equivalence.

In fact, for any behavioral equivalence, one of the most frequently-asked problems is whether or not it can be preserved under some frequently-used operators, such as prefixing, choice, parallel composition, etc., in classical process algebraic languages like CCS [8], CSP [6] and ACP [9]. Generally, there exist two ways to deal with this problem: The first one is to prove the congruence properties of these operators one by one. It is a straightforward and intuitive way, but may be somewhat clumsy. The second one is to pursue a rule format for this specified behavioral equivalence. And the given behavioral equivalence can be preserved under any operators in this format.

However, we have noticed that equivalences in strong notion, such as strong bisimulation and decorated trace semantics, attracted much more attention equivalences in weak notion such as weak bisimulation and testing theory did. In fact, almost all the classical strong equivalences have found their corresponding rule formats, but much less works have been done on the rule formats of weak equivalences, especially on the rule formats of the equivalences in testing theoretical notions. And more specifically, no rule formats have been presented to be congruence formats for the weak readiness equivalence or the weak possible future equivalence. The difference may exist in the increasing complexity after introducing τ transitions by weak equivalences. The aim of our paper is to make a progress along this direction.

Recently, another kind of proof technique on congruence formats based on decomposing of the subclass of modal formulas for some given equivalences has been studied [23, 24, 25]. Within their works, (bi)simulation-like equivalences and decorated trace preorders have found their corresponding congruence formats. These formats are sub-formats of ready simulation format, i.e., ntyft/ntyxt format without lookahead. However, the problem whether this proof technique can be fit into our works aiming at weak readiness equivalence and weak possible future equivalence is still open.

Acknowledgement The authors would like to thank the anonymous referees for their meaningful comments and suggestions.

References

[1] R.J. van Glabbeek (2001). The Linear Time - Branching Time Spectrum I: The Semantics of Concrete, Sequential Processes. In J.A. Bergstra, A. Ponse, and S.A. Smolka (eds), Handbook of Process Algebra, chapter 1, pages 3-100. Elsevier.

[2] R.J. van Glabbeek (1993). The Linear Time - Branching Time Spectrum II: The semantics of sequential systems with silent moves. In E. Best (eds), Concur'93, LNCS 715, pages 66-81. Springer-Verlag, Berlin.

[3] E.-R. Olderog and C.A.R. Hoare (1986). Specification-oriented semantics for communicating processes. Acta Informatica 23, pages 9-66.

[4] W.C. Rounds and S.D. Brooks (1981). Possible futures, acceptances, refusals and communicating processes. In Proceedings of the 22th Annual Symposium on Foundations of Computer Science, Nashville, Tennessee, pages 140-149, IEEE, New York.

[5] P.Y.A. Ryan (2001). Mathematical Models of Computer Security. In R. Focardi and R. Gorrieri (eds), Foundations of Security Analysis and Design, Lecture Notes in Computer Science 2171, pages 1-62. Springer-Verlag, Berlin.

[6] C.A.R. Hoare (1985), Communicating Sequential Processes, Prentice-Hall, Englewood Cliffs, NJ.

[7] G.D. Plotkin (2004). A Structural Approach to Operational Semantics. The Journal of Logic and Algebraic Programming 60-61, pages 17-139.

[8] R. Milner (1989). Communication and Concurrency. Prentice-Hall.

[9] J.C.M. Baeten and W.P. Weijland (1990). Process Algebra. volume 18 of Cambridge Tracts in Theoretical Computer Science, Cambridge University Press.

[10] J.F. Groote and F.W. Waandrager (1992). Structural Operational Semantics and Bisimulation as a Congruence. Information and Computation 100(2), pages 202-260.

[11] S. Burris and H.P. Sankappanavar (1981). A Course in Universal Algebra, Graduate Texts in Mathematics, vol. 78, Springer-Verlag, Berlin.

[12] B. Bloom, S. Istrail and A. R. Meyer (1995). Bisimualtion can't be Traced. Journal of the ACM 42(1), pages 232-268.

[13] J.F. Groote (1993). Transition System Specifications with Negative Premises. Theoretical Computer Science 118, pages 263-299.

[14] R.D. Simone (1985). Higher-level synchronising devices in Meije-SCCS. Theoretical Computer Science 37, pages 245-267.

[15] A. Rensink, W. Vogler (2007). Fair testing. Information and Computation 205(2), pages 125-198.

[16] L. Aceto, W.J. Fokkink and C. Verhoef (2001). Structural Operational Semantics. In J.A. Bergstra, A. Ponse and S.A. Smolka, (eds), Handbook of Process Algebra, Chapter 3, pages 197-292. Elsevier.

[17] M.R. Mousavi, M.A. Reniers, J.F. Groote (2007). SOS formats and meta-theory: 20 years after. Theoretical Computer Science 373, pages 238-272.

[18] W.J. Fokkink (2000), Rooted branching bisimulation as a congruence, Journal of Computer and System Sciences 60(1), pages 13-37.

[19] B. Bloom (1995). Structural operational semantics for weak bisimulations. Theoretical Computer Science 146, pages 27-68.

[20] R.J. van Glabbeek (2005), On Cool Congruence Formats for Weak Bisimulations. In D.V. Hung and M. wirsing (eds), In Proceedings of the 2th International Colloquium on Theoretical Aspects of Computing (ICTAC2005), Hanoi, Vietnam, October, 2005. Lecture Notes in Computer Science 3722, pages 331-346. Springer-Verlag, Berlin.

[21] I. Ulidowski (2000), Finite Axiom Systems for Testing Preorder and De Simone Process Languages. Theoretical computer Science 239(1), pages 97-139.

[22] B. Bloom (1990). Ready Simulation, Bisimulation, and the Semantics of CCS-Like Languages. PhD thesis, MIT.

[23] B. Bloom, W.J. Fokkink, and R.J. van Glabbeek (2004). Precongruence Formats for Decorated Trace Semantics. ACM Transactions on Computational Logic 5(1), pages 26-78.

[24] W.J. Fokkink, R.J. van Glabbeek and P. de Wind (2005). Divide and Congruence Applied to Eta-Bisimulation. In Proceedings of the Second Workshop on Structural Operational Semantics (SOS 2005), Lisbon, Portugal, July, 2005. Electronic Notes in Theoretical Computer Science 156(1), pages 97-113. Elsevier.

[25] W.J. Fokkink, R.J. van Glabbeek, P. de Wind (2005), Divide and congruence: from decomposition of modalities to preservation of branching bisimulation, In Proceedings of Fourth International Symposium on Formal Methods for Components and Objects (FMCO'05), Amsterdam, November, 2005. Lecture Notes in Computer Science 4111, pages 195-218. Springer-Verlag, Berlin.