通信プロセスの弱線型意味論に対する完全抽象合成的モデル

堀田 英一
NTT ソフトウェア研究所
〒180, 東京都, 武蔵野市, 緑町, 3-9-11

通信プロセスを記述するためのある言語 $\mathcal{L}$ の意味論を考察する. $\mathcal{L}$ は CCS の サブ・セットであり, 動作前置, 非決定的選択, 並行合成, 再帰法を含む. 最初に, 操作的意味論 $\mathcal{O}_{\mathrm{WL}}$ を, Plotkin のスタイル で定義する. この意味論はプログラムの意味が, その実行可能なイベント列の集合からなるという意味で線型であり, またそのイベント列は, (環境からは不可視な) 内部イベントを抽象化して得られるものであるという意味で弱型である. 次に合成的モデル $\mathcal{C}_{\mathrm{RF}}$ を定義し, $\mathcal{C}_{\mathrm{RF}}$ の $\mathcal{O}_{\mathrm{WL}}$ に対する **完全抽象性**, 即ち次式が成り立つことを示す:

$$\mathcal{C}_{\mathrm{RF}}(s_1) = \mathcal{C}_{\mathrm{RF}}(s_2) \Leftrightarrow \forall C[C \text{ は } \mathcal{L} \text{ の 文脈} \Rightarrow \mathcal{O}_{\mathrm{WL}}(C[s_1]) = \mathcal{O}_{\mathrm{WL}}(C[s_2])\ ].$$

A Fully Abstract Model for Communicating Processes with respect to Weak Linear Semantics

Eiichi Horita
NTT Software Laboratories
3-9-11 Midori-Cho, Musashino-Shi, Tokyo 180, Japan

The semantics of a language $\mathcal{L}$ for communicating processes is investigated. It contains *action prefixing*, *nondeterministic choice*, *parallel composition*, and *recursion*. A Plotkin-style operational semantics $\mathcal{O}_{\mathrm{WL}}$ is defined. This semantics is *linear* in that the meaning of each program in $\mathcal{O}_{\mathrm{WL}}$ is a set of *event sequences* the program may perform, and is *weak* in that the event sequences are obtained by ignoring *internal moves*. Then, a compositional model $\mathcal{C}_{\mathrm{RF}}$ is proposed, and its *full abstractness*, as expressed in the following, is established:

$$\mathcal{C}_{\mathrm{RF}}(s_1) = \mathcal{C}_{\mathrm{RF}}(s_2) \Leftrightarrow \forall C[C \text{ is a } \textit{context} \text{ of } \mathcal{L} \Rightarrow \mathcal{O}_{\mathrm{WL}}(C[s_1]) = \mathcal{O}_{\mathrm{WL}}(C[s_2])\ ].$$

1 Introduction

The semantics of a language $\mathcal{L}$ for communicating processes is investigated. The language $\mathcal{L}$ is a subset of CCS ([Mil 80]) containing *action prefixing, nondeterministic choice, parallel composition*, and a form of *recursion.*

First, an operational semantics $\mathcal{O}_{\rm WL}$ of $\mathcal{L}$ is defined in terms of a labeled transition system, in the style of Plotkin ([Plo 81]). This semantics is *linear* in that the meaning of each statement $s \in \mathcal{L}$ in this semantics is a set of *event sequences*, which the process represented by s may perform; it is *weak* in that the event sequences are obtained by ignoring *internal moves* (denoted by τ in [Mil 80]) invisible to its environment.

Next, a compositional model $\mathcal{C}_{\rm RF}$ is proposed, which is a variant of the *failures model* proposed by Brookes, Hoare, and Roscoe ([BHR 84]) and later improved ([BR 84]). It is shown that $\mathcal{C}_{\rm RF}$ is *fully abstract* w.r.t. $\mathcal{O}_{\rm WL}$. That is, $\mathcal{C}_{\rm RF}$ is the most abstract compositional model which is correct w.r.t. $\mathcal{O}_{\rm WL}$. Equivalently, one can obtain the following for every $s_1, s_2 \in \mathcal{L}$:

$$\mathcal{C}_{\rm RF}(s_1) = \mathcal{C}_{\rm RF}(s_2) \Leftrightarrow \forall C[C \text{ is a } \textit{context} \text{ of } \mathcal{L} \Rightarrow \mathcal{O}_{\rm WL}(C[s_1]) = \mathcal{O}_{\rm WL}(C[s_2])\ ]. \qquad (1)$$

A similar full abstractness result has been established by Bergstra, Klop, and Olderog for a language with no recursion and internal moves ([BKO 88]). Rutten discussed the semantics of a language similar to $\mathcal{L}$, in the framework of complete metric spaces, and showed, along the lines of the proof of a similar statement in [BKO 88], that the failures model is fully abstract with respect to a *strong linear semantics* $\mathcal{O}_{\rm L}$ ([Rut 89]), where $\mathcal{O}_{\rm L}$ is *strong* in that it does not abstract from internal moves. The result described above is an extension of the result in [BKO 88] to a language with recursion and internal moves; it is also an extension of the result in [Rut 89] to the case of weak semantics instead of strong semantics.

The *full abstractness problem* for programming languages was first raised by Milner ([Mil 73]). In general, a fully abstract model for a given language w.r.t. a given operational semantics $\mathcal{O}$ is the most desirable one from a viewpoint associated with $\mathcal{O}$. In particular, the fully abstract model $\mathcal{C}_{\rm RF}$ is the most desirable one from the following viewpoint: In some practical areas, the most interesting characteristic of a (software or hardware) system is the set of (visible) event sequences which the system may perform. One cannot define, however, a compositional model consisting of such sets of sequences in the concurrent setting, as was shown, e.g., by Milner (cf. [Mil 80], § 1.2), which is also exhibited in Example 1 in the present setting. Compositionality, in turn, is needed for the *stepwise* definition of program meanings. In other words, the meaning of a composite statement needs to be defined in terms of the meanings of its components. It is also necessary to treat systems as *modules*, i.e., to make it possible for two equivalent systems, A and B, to substitute A for B within composite systems, without affecting the overall meaning. Thus some *extra* information needs to be involved to construct a compositional model. However, it is desirable for the extra information to be *minimum* so as not to bring about inessential details. The fully abstract compositional model $\mathcal{C}_{\rm RF}$ meets these requirements.

Although the model $\mathcal{C}_{\rm RF}$ is compositional and the meaning of each recursive program under $\mathcal{C}_{\rm RF}$ is a fixed point of the associated function (the interpretation of the body of its defining equation), it is not *denotational* in the framework of *complete partially ordered sets*, where the meaning of a recursive program is defined as the *least* fixed point of the associated function. Furthermore, $\mathcal{C}_{\rm RF}$ is not denotational in the framework of *complete metric spaces*, where the meaning of a recursive program is defined as the *unique* fixed point of the associated contraction (cf. [BZ 82]). Such an order-theoretic or metric topological construction of $\mathcal{C}_{\rm RF}$ remains for future study (cf. § 8). Note that such a denotational construction of $\mathcal{C}_{\rm RF}$ does not necessarily exist as was shown in [AP 86]. In [HP 79], a fully abstract model for a parallel language was constructed in an order-theoretic framework. However, the concurrency treated there is different from the one treated here, because the language in [HP 79] includes *coroutine* construct as well as the usual interleaving. The characterization of $\mathcal{C}_{\rm RF}$ in this paper as a fully abstract model w.r.t. $\mathcal{O}_{\rm WL}$ is analogous to Milner's characterization of the so-called *observation congruence* in [Mil 83] and [Mil 85].

Brookes considered the relation between two models of concurrent behavior, Milner's *synchronization trees* for CCS, and the *failures model* of TCSP ([Bro 83]), where only finite processes defined without recursion were dealt with. Since this paper investigates the relation between labeled transition systems, which are mapped into synchronization trees by a natural translation (and vice versa), and a variant of the failures model for infinite processes, the connection described in this paper is regarded as an extension of the one in [Bro 83] to the case of infinite processes. Our model $\mathcal{C}_{\rm RF}$ differs from the original failures model in [BHR 84] even for finite processes, because we treat '+', *alternative composition* of CCS, and the original model is not a congruence w.r.t. this operator even for finite processes. In [Bro 83], this modification was not needed, because only TCSP operators, which do not include '+', were treated there.

2 Notation and Mathematical Preliminaries

The underlying structures of the models introduced in § 4 and § 5, are domains of (finite or infinite) sequences of some elements. Sequences are treated in the standard manner in set theory, using the notations below (cf., e.g., [Kun 80]).

The phrase "let $(x \in)X$ be $\cdots$" introduces a set X with variable x ranging over X.

Notation 1

(1) The standard λ-notation is used for denoting functions: For a set A, a variable x, and an expression $E(x)$, the expression $(\lambda x \in A : E(x))$ denotes the function which maps $x \in A$ to $E(x)$.

(2) For a set X, the cardinality of X is denoted by $\sharp(X)$. For two sets X and Y, the set of functions from X to Y is denoted by $(X \to Y)$ or by Y^X. The set of natural numbers is denoted by ω. Let $\bar{\omega} = \omega \setminus \{0\}$. Each number $n \in \omega$ is identified with the set $\{i \in \omega : 0 \le i < n\}$ as usual in set theory. For $n \in \omega$, let $\bar{n} = \{m \in \omega : 1 \le m \le n\}$. ∎

Notation 2

(1) The empty sequence is denoted by ϵ.

(2) For a set A, the set of finite sequences of elements of A is denoted by $A^{<\omega}$, and the set of nonempty finite sequences of elements of A is denoted by A^{+}. The set of finite or infinite (with length ω) sequences of elements of A is denoted by $A^{\le\omega}$.
For $a \in A$, the sequence (a) consisting only of a is sometimes denoted by a.

(3) Each sequence $q \in A^{\le\omega}$ is regarded as a *function* whose domain is a member of $\omega \cup \{\omega\}$. Thus, referring to its *length* as $\mathrm{dom}(q)$, one has $q = (\lambda i \in \mathrm{dom}(q) : q(i))$. For $a \in A$ and $\nu \in \omega \cup \{\omega\}$, let $a^{\nu} = (\lambda i \in \nu : a)$.

(4) For $q \in (A^{\le\omega} \setminus \{\epsilon\})$, $\mathrm{rest}(q)$ denotes the unique sequence $\tilde{q} \in A^{\le\omega}$ such that there is an isomorphism $\phi : \mathrm{dom}(\tilde{q}) \to (\mathrm{dom}(q) \setminus \{0\})$ satisfying $\forall i \in \mathrm{dom}(q)[\ \tilde{q}(i) = q(\phi(i))]$.

(5) For $q_1 \in A^{<\omega}$ and $q_2 \in A^{\le\omega}$, let $q_1 \cdot q_2$ denote the concatenation of q_1 and q_2. Also, for $p_1 \subseteq A^{<\omega}$ and $p_2 \subseteq A^{\le\omega}$, let $p_1 \cdot p_2 = \{q_1 \cdot q_2 : q_1 \in p_1 \wedge q_2 \in p_2\}$.

(6) For $p \subseteq A^{\le\omega}$ and $q \in A^{<\omega}$, let $p[q] = \{\tilde{q} \in A^{\le\omega} : q \cdot \tilde{q} \in p\}$.

(7) For $q_1, q_2 \in A^{\le\omega}$, let us write $q_1 \le_p q_2$ to denote that q_1 is a *prefix* of q_2. The relation $\le_p$ is known as the *prefix ordering*. ∎

The notion of a *homomorphism* is defined below; it is used to define the *merging* of two sequences in § 5.2.

Definition 1 Let A and B be sets. A function $h : A^{\le\omega} \to B^{\le\omega}$ is called a *homomorphism* iff $h[A^{<\omega}] \subseteq B^{<\omega}$ and $\forall q_1 \in A^{<\omega}, \forall q_2 \in A^{\le\omega}[\ h(q_1 \cdot q_2) = h(q_1) \cdot h(q_2)\ ]$. ∎

It easy to see that a homomorphism $h : A^{\le\omega} \to B^{\le\omega}$ is determined by the values $h(a)$ $(a \in A)$.

3 A Language $\mathcal{L}$ for Communicating Processes

In this section, a language $\mathcal{L}$ for communicating processes is defined. This is a subset of CCS ([Mil 80]) containing *action prefixing*, *nondeterministic choice*, *parallel composition*, and a form of *recursion*.

Definition 2 Let $(X \in)\mathcal{V}$ be a set of *statement variables*. First, a language $(S \in)$ $\mathcal{L}[\mathcal{V}]$ with general recursion is defined as follows:

$$S ::= D \mid (c; S) \mid (\tau; S) \mid (S_1 + S_2) \mid (S_1 \parallel S_2) \mid X \mid \mu X(S), \tag{2}$$

where D, τ, $+$, and $\parallel$ represent *deadlock*, the *internal move*, the *alternative composition*, and the *parallel composition*, respectively; c ranges over $\mathbf{C}$, a set of *communication actions*. For $S \in \mathcal{L}[\mathcal{V}]$, let $\mathrm{FV}(S)$ be the set of *free variables* contained in S. Intuitively $\mu X(S)$ stands for a *solution* of the equation $X = S$. Syntactically the prefix "μX" *binds* each variable X, as "λx" in λ-notation.

Next, $(S \in)$ $\widehat{\mathcal{L}}[\mathcal{V}]$, a sublanguage of $\mathcal{L}[\mathcal{V}]$ with a restriction on recursion, is defined to be the set of $S \in \mathcal{L}[\mathcal{V}]$ satisfying the following restriction:

$$\text{If } \mu X(S') \text{ is a subexpression of } S, \text{ then } \mathrm{FV}(S') \subseteq \{X\}. \tag{3}$$

For $\mathcal{V}' \subseteq \mathcal{V}$, let $\mathcal{L}[\mathcal{V}'] = \{S \in \mathcal{L}[\mathcal{V}] :\ \mathrm{FV}(S) \subseteq \mathcal{V}'\}$. We write $\mathcal{L}[X]$ for $\mathcal{L}[\{X\}]$ $(X \in \mathcal{V})$. Finally, let $(s \in)$ $\mathcal{L} = \mathcal{L}[\emptyset]$. ∎

The reason why the restriction on recursion is imposed is stated in the following: For the restricted language $\widehat{\mathcal{L}}[\mathcal{V}]$, the *correctness* of the model $\mathcal{C}_{\mathrm{RF}}$ (to be presented in § 5) w.r.t. the weak linear operational semantics, follows immediately from its compositionality. However, this is not the case for the unrestricted language $\mathcal{L}[\mathcal{V}]$: The proof of the correctness for $\mathcal{L}[\mathcal{V}]$ requires a considerable amount of work, in addition to the compositionality. Since this paper focuses on the full abstractness of $\mathcal{C}_{\mathrm{RF}}$, which can be demonstrated by discussing only $\widehat{\mathcal{L}}[\mathcal{V}]$, we first establish the full abstractness of $\mathcal{C}_{\mathrm{RF}}$ for $\widehat{\mathcal{L}}[\mathcal{V}]$. The proof of the correctness of $\mathcal{C}_{\mathrm{RF}}$ for $\mathcal{L}[\mathcal{V}]$ is outlined in the Appendix.

Note that, once the correctness for $\mathcal{L}[\mathcal{V}]$ has been established, the full abstractness for $\mathcal{L}[\mathcal{V}]$ follows in the same way as for $\widehat{\mathcal{L}}[\mathcal{V}]$.

Notation 3 For $S, S' \in \mathcal{L}[\mathcal{V}]$, we write $S \equiv S'$, to denote that S and S' are syntactically identical.
For $S, S' \in \mathcal{L}[\mathcal{V}]$ and $X \in \mathcal{V}$, we denote by $S[S'/X]$ the result of substituting S' for all free occurrences of X in S. ∎

4 Weak Linear Semantics $\mathcal{O}_{\mathrm{WL}}$ for $\mathcal{L}$

The weak linear operational semantics $\mathcal{O}_{\mathrm{WL}}$ of $\mathcal{L}$ is defined as usual in the style of Plotkin ([Plo 81]). Here "WL" stands for Weak Linear Model. For the definition, some preliminaries are needed.

Definition 3

(1) A bijection $\bar{\cdot} : \mathbf{C} \to \mathbf{C}$ is assumed to be given such that for every $c \in \mathbf{C}$, $\bar{\bar{c}} = c$.

(2) Let $\mathbf{A} = \mathbf{C} \cup \{\tau\}$.

(3) A transition relation $\to \subseteq (\mathcal{L} \times \mathbf{A} \times \mathcal{L})$ is defined as the smallest set satisfying the following rules. For $s_1, s_2 \in \mathcal{L}$, $a \in \mathbf{A}$, we write $s_1 \xrightarrow{a} s_2$ for $(s_1, a, s_2) \in \to$.

(i) $(a; s) \xrightarrow{a} s.$

(ii)

$$\frac{s_1 \xrightarrow{a} s_1'}{\begin{array}{l}(s_1 + s_2) \xrightarrow{a} s_1' \\ (s_2 + s_1) \xrightarrow{a} s_1'\end{array}}$$

(iii)

$$\frac{s_1 \xrightarrow{a} s_1'}{\begin{array}{l}(s_1 \parallel s_2) \xrightarrow{a} s_1' \parallel s_2 \\ (s_2 \parallel s_1) \xrightarrow{a} s_2 \parallel s_1'\end{array}}$$

(iv)

$$\frac{s_1 \xrightarrow{c} s_1',\, s_2 \xrightarrow{\bar{c}} s_2}{s_1 \parallel s_2 \xrightarrow{\tau} s_1' \parallel s_2'} \quad (c \in \mathbf{C})$$

(v)

$$\frac{S[\mu X(S)/X] \xrightarrow{a} s'}{\mu X(S) \xrightarrow{a} s'}$$

This rule is called the *recursion rule*.

(4) Then for $w \in \mathbf{A}^{<\omega}$, a binary relation $\xrightarrow{w}_*$ is defined recursively by:

(i) $s \xrightarrow{\epsilon}_* s,$

(ii) $s \xrightarrow{(a)\cdot w}_* s'$ iff $\exists s''[\, s \xrightarrow{a} s'' \wedge s'' \xrightarrow{w}_* s' \,]$.

(5) For $w \in \mathbf{C}^{<\omega}$, a binary relation $\xRightarrow{w}$ is defined by:

$$s \xRightarrow{w} s' \text{ iff } \exists w' \in \mathbf{A}^{<\omega}[\, (w' \setminus \tau) = w \wedge s \xrightarrow{w'}_* s' \,],$$

where $(w' \setminus \tau)$ is the result of erasing τ's in w'.

(6) For $s \in \mathcal{L}$, let $\mathrm{act}(s) = \{a \in \mathbf{A} : \exists s'[\, s \xrightarrow{a} s' \,]\}$. ∎

By means of these notions, the weak linear operational semantics $\mathcal{O}_{\mathrm{WL}}$ is defined as follows.

Definition 4

(1) The domain of $\mathcal{O}_{\mathrm{WL}}$, written $\mathbf{D}(\mathcal{O}_{\mathrm{WL}})$, is defined by:

$$\mathbf{D}(\mathcal{O}_{\mathrm{WL}}) = (\mathbf{C}^{<\omega} \cdot \{(\delta), (\bot)\}) \cup \mathbf{C}^{\omega},$$

where δ and $\bot$ are distinct symbols representing *deadlock* and divergence, respectively.

(2) For $s \in \mathcal{L}$, let $\mathrm{Traces}_\delta(s)$ be the set of *traces ending with deadlock*, $\mathrm{Inf}(s)$ the set of *infinite paths*, and $\mathrm{Div}(s)$ the set of *divergences*. Here $\mathrm{Traces}_\delta(s)$, $\mathrm{Inf}(s)$, and $\mathrm{Div}(s)$ are defined as follows:

$$\mathrm{Traces}_\delta(s) = \{w \cdot (\delta) : w \in \mathbf{C}^{<\omega} \wedge \exists s'[\, s \xRightarrow{w} s' \wedge \mathrm{act}(s') \subseteq \mathbf{C} \,]\}, \tag{4}$$

$$\mathrm{Inf}(s) = \{(c_n)_{n\in\omega} : \exists (s_n)_{n\in\omega}[\, s_0 = s \wedge \forall n[\, s_n \xRightarrow{c_n} s_{n+1} \,]]\}, \tag{5}$$

$$\mathrm{Div}(s) = \{w \cdot (\perp) : \exists s', \exists (s_n)_{n\in\omega}[\ s \overset{w}{\Longrightarrow} s' \wedge s_0 = s' \wedge \forall n[\ s_n \overset{\tau}{\longrightarrow} s_{n+1}\ ]]\}. \tag{6}$$

(3) For $s \in \mathcal{L}$, let

$$\mathcal{O}_{\mathrm{WL}}(s) = \mathrm{Trace}_\delta(s) \cup \mathrm{Inf}(s) \cup \mathrm{Div}(s). \quad \blacksquare$$

As stated in the introduction, $\mathcal{O}_{\mathrm{WL}}$ is not compositional as is exhibited in the following example.

Example 1 Let $s_1 \equiv (c_0; c_1; D) + (c_0; c_2; D)$, $s_2 \equiv c_0; ((c_1; D) + (c_2; D))$. Then, $\mathcal{O}_{\mathrm{WL}}(s_1) = \mathcal{O}_{\mathrm{WL}}(s_2)$. However, putting $s \equiv \overline{c_1}; \mu X(\tau; X)$, one has

$(c_0, \delta) \in \mathcal{O}_{\mathrm{WL}}(s_1 \parallel s) \setminus \mathcal{O}_{\mathrm{WL}}(s_2 \parallel s)$, and therefore, $\mathcal{O}_{\mathrm{WL}}(s_1 \parallel s) \neq \mathcal{O}_{\mathrm{WL}}(s_2 \parallel s)$. $\blacksquare$

5 Compositional Model $\mathcal{C}_{\mathrm{RF}}$ for $\mathcal{L}$

In this section a compositional model $\mathcal{C}_{\mathrm{RF}}$ for $\mathcal{L}$ is defined. It is a mild variant of the failures model of [BHR 84] and can be shown to be a fully abstract compositional model w.r.t. the operational semantics $\mathcal{O}_{\mathrm{WL}}$. Here "RF" stands for *Rooted Failures Model.*[1]

5.1 Definition of $\mathcal{C}_{\mathrm{RF}}$

First, the domain of $\mathcal{C}_{\mathrm{RF}}$, written $\mathbf{D}(\mathcal{C}_{\mathrm{RF}})$, is defined as follows:

Definition 5 Let us use Γ as a variable raging over $\wp(\mathbf{C})$. First, let

$$\mathbf{B}(\mathcal{C}_{\mathrm{RF}}) = (\mathbf{C}^{<\omega} \cdot \{(\langle \delta, \Gamma \rangle) : \Gamma \subseteq \mathbf{C}\}) \cup \{(\langle \hat{\delta}, \Gamma \rangle) : \Gamma \subseteq \mathbf{C}\} \cup \mathbf{C}^{\omega} \cup (\mathbf{C}^{<\omega} \cdot \{(\perp)\}).$$

By means of this, $\mathbf{D}(\mathcal{C}_{\mathrm{RF}})$ is defined by: $\mathbf{D}(\mathcal{C}_{\mathrm{RF}}) = \wp(\mathbf{B}(\mathcal{C}_{\mathrm{RF}}))$.
For $p \in \mathbf{D}(\mathcal{C}_{\mathrm{RF}})$, let

$$\mathcal{F}^\circ(p) = p \cap (\mathbf{C}^{<\omega} \cdot \{\langle \delta, \Gamma \rangle : \Gamma \subseteq \mathbf{C}\}),\ \mathcal{R}^\circ(p) = p \cap \{(\langle \hat{\delta}, \Gamma \rangle) : \Gamma \subseteq \mathbf{C}\},$$
$$\mathrm{Inf}^\circ(p) = p \cap (\mathbf{C}^{\omega}),\ \mathrm{Div}^\circ(p) = p \cap (\mathbf{C}^{<\omega} \cdot \{(\perp)\}). \quad \blacksquare$$

The compositional model $\mathcal{C}_{\mathrm{RF}}(s) : \mathcal{L} \to \mathbf{D}(\mathcal{C}_{\mathrm{RF}})$ is defined by:

Definition 6

(1) For $s \in \mathcal{L}$, let $\mathcal{F}(s)$ be the set of *failures* in the usual sense, $\mathcal{R}(s)$ the set of refusals of s (not of some s' such that $s \overset{\epsilon}{\Longrightarrow} s'$).
Here $\mathcal{F}(s)$ and $\mathcal{R}(s)$ are defined as follows:

$$\mathcal{F}(s) = \{w \cdot (\langle \delta, \Gamma \rangle) : \exists s'[\ s \overset{w}{\Longrightarrow} s' \wedge \mathrm{act}(s') \subseteq \mathbf{C} \wedge \Gamma \subseteq \mathbf{C} \wedge \Gamma \cap \mathrm{act}(s') = \emptyset\ ]\}, \tag{7}$$

$$\mathcal{R}(s) = \{(\langle \hat{\delta}, \Gamma \rangle) : (\mathrm{act}(s) \subseteq \mathbf{C}) \wedge \Gamma \subseteq \mathbf{C} \wedge \Gamma \cap \mathrm{act}(s) = \emptyset\}. \tag{8}$$

(2) For $s \in \mathcal{L}$, let

$$\mathcal{C}_{\mathrm{RF}}(s) = \mathcal{F}(s) \cup \mathcal{R}(s) \cup \mathrm{Inf}(s) \cup \mathrm{Div}(s). \quad \blacksquare \tag{9}$$

Note that from $\mathcal{C}_{\mathrm{RF}}(s)$, each of $\mathcal{F}(s)$, $\mathcal{R}(s)$, $\mathrm{Inf}(s)$, $\mathrm{Div}(s)$ is represented as follows: $\mathcal{F}(s) = \mathcal{F}^\circ(\mathcal{C}_{\mathrm{RF}}(s))$, $\mathcal{R}(s) = \mathcal{R}^\circ(\mathcal{C}_{\mathrm{RF}}(s))$, $\mathrm{Inf}(s) = \mathrm{Inf}^\circ(\mathcal{C}_{\mathrm{RF}}(s))$, $\mathrm{Div}(s) = \mathrm{Div}^\circ(\mathcal{C}_{\mathrm{RF}}(s))$.

Remark 1

(1) The definition of $\mathcal{C}_{\mathrm{RF}}$ is nonstandard in the sense that it contains both failures and refusals. The refusal part $\mathcal{R}(s)$ is added to distinguish two statements such that their failure sets are the same but their operational meanings are different in some context of $\mathcal{L}$. Let $\psi : \{(\langle \hat{\delta}, \Gamma \rangle) : \Gamma \subseteq \mathbf{C}\} \to \{(\langle \delta, \Gamma \rangle) : \Gamma \subseteq \mathbf{C}\}$ be defined by $\psi((\langle \hat{\delta}, \Gamma \rangle)) = (\langle \delta, \Gamma \rangle)$. Then for every $s \in \mathcal{L}$, $\mathcal{R}(s)$ is embedded into $\mathcal{F}(s)$, i.e., $\psi[\mathcal{R}(s)] \subseteq \mathcal{F}(s)$.
Thus, for every element $(\langle \hat{\delta}, \Gamma \rangle) \in \mathcal{R}(s)$, a copy $(\langle \delta, \Gamma \rangle) \in \mathcal{F}(s)$ exists. The difference between $(\langle \hat{\delta}, \Gamma \rangle)$ and $(\langle \delta, \Gamma \rangle)$ is that $(\langle \hat{\delta}, \Gamma \rangle)$ is an immediate refusal of s itself (not of s' such that $s \overset{\epsilon}{\Longrightarrow} s'$), while $(\langle \delta, \Gamma \rangle)$ is a refusal of some s' such that $s \overset{\epsilon}{\Longrightarrow} s'$ (s' may be s itself by the definition of $\overset{\epsilon}{\Longrightarrow}$). In other words, $(\langle \hat{\delta}, \Gamma \rangle)$ must stem from the *root* of the transition tree of s, while $(\langle \delta, \Gamma \rangle)$ may not.

(2) There are alternative formulations for $\mathcal{C}_{\mathrm{RF}}$. For example, let us define $\mathcal{R}'$ as follows: For $s \in \mathcal{L}$,

$$\mathcal{R}'(s) = \begin{cases} \{\hat{\delta}\} & \text{if } \mathrm{act}(s) \subseteq \mathbf{C}, \\ \emptyset & \text{otherwise.} \end{cases}$$

Then it is easy to see that for every $s_1, s_2 \in \mathcal{L}$, the following holds:

$$\mathcal{F}(s_1) \cup \mathcal{R}(s_1) = \mathcal{F}(s_1) \cup \mathcal{R}(s_1) \Leftrightarrow \mathcal{F}(s_1) \cup \mathcal{R}'(s_1) = \mathcal{F}(s_1) \cup \mathcal{R}'(s_1).$$

Thus the part $\mathcal{R}(s)$ can be replaced by a 1-bit piece of information $\mathcal{R}'(s)$. We prefer the present formulation for the convenience of the definition of semantic operations in § 5.2 and of the correctness proof in the Appendix. $\blacksquare$

[1] The terminology is partly borrowed from [BK 85], where the notion of *rooted τ-bisimulation* is proposed.

5.2 Compositionality of $\mathcal{C}_{\mathrm{RF}}$

It can be shown that $\mathcal{C}_{\mathrm{RF}}$ is a *congruence* w.r.t. all operators of $\mathcal{L}$, and therefore, $\mathcal{C}_{\mathrm{RF}}$ is *compositional.* For this purpose, semantics operations corresponding to the syntactic operators of $\mathcal{L}$ are defined.

First, for each $a \in \mathbf{A}$, a unary semantic operation prefix_a corresponding to syntactical prefixing of a is defined.

Definition 7

(1) Let $c \in \mathbf{C}$. First, two auxiliary operations $\mathcal{F}_c$, $\mathcal{R}_c$ are defined as follows: For $p \in \mathbf{D}(\mathcal{C}_{\mathrm{RF}})$,

$$\mathcal{F}_c(p) = \{(\langle\delta,\Gamma\rangle) : \ \Gamma \subseteq \mathbf{C} \wedge c \notin \Gamma\} \ \cup \ (c) \cdot \mathcal{F}^\circ(p),$$
$$\mathcal{R}_c(p) = \{(\langle\hat{\delta},\Gamma\rangle) : \ \Gamma \subseteq \mathbf{C} \wedge c \notin \Gamma\}.$$

From these, prefix_c is defined as follows: For $p \in \mathbf{D}(\mathcal{C}_{\mathrm{RF}})$,

$$\mathrm{prefix}_c(p) = \mathcal{F}_c(p) \cup \mathcal{R}_c(p) \ \cup \ (c) \cdot \mathrm{Inf}^\circ(p) \ \cup \ (c) \cdot \mathrm{Div}^\circ(p).$$

(2) A unary operation prefix_τ is defined as follows: For $p \in \mathbf{D}(\mathcal{C}_{\mathrm{RF}})$,

$$\mathrm{prefix}_\tau(p) = p \setminus \mathcal{R}^\circ(p). \quad \blacksquare$$

Next, a binary semantic operation $\tilde{+}$ corresponding to '+' is defined.

Definition 8 With an auxiliary operation $\mathcal{F}_+$, $\tilde{+}$ is defined as follows: For $p_1, p_2 \in \mathbf{D}(\mathcal{C}_{\mathrm{RF}})$,

$$\mathcal{F}_+(p_1,p_2) = (\mathcal{F}^\circ(p_1) \setminus \psi[\mathcal{R}^\circ(p_1)]) \cup (\mathcal{F}^\circ(p_2) \setminus \psi[\mathcal{R}^\circ(p_2)]) \cup \psi[\mathcal{R}^\circ(p_1) \cap \mathcal{R}^\circ(p_2)],$$
$$p_1 \tilde{+} p_2 = \mathcal{F}_+(p_1,p_2) \cup (\mathcal{R}^\circ(p_1) \cap \mathcal{R}^\circ(p_2)) \cup (\mathrm{Inf}^\circ(p_1) \cup \mathrm{Inf}^\circ(p_2)) \cup (\mathrm{Div}^\circ(p_1) \cup \mathrm{Div}^\circ(p_2)). \quad \blacksquare$$

Note that the failures-part of $(s_1 + s_2)$ is composed of the failures- and refusals-parts of s_1 and s_2, while the refusals-part of $(s_1 + s_2)$ is composed only of the refusals-parts of s_1 and s_2.

Finally, a binary semantic operation $\tilde{\|}$ corresponding to '$\|$' is defined. As a preliminary to the definition, a function $\mathrm{merge}_w : (\mathbf{A}^{\le\omega} \times \mathbf{A}^{\le\omega}) \to \wp(\mathbf{C}^{\le\omega})$ is defined by:

Definition 9 Let $q_1, q_2 \in \mathbf{A}^{\le\omega}$.

(1) First, the set of merged sequences of q_1 and q_2 with extra information on the origin of its elements, written $\mathrm{merge}^*(q_1,q_2)$, is defined. Let L, R, S be distinct symbols standing for 'Left', 'Right', 'Synchronization', respectively; put

$$\mathbf{R} = \{\rho \in (\{L,R,S\} \times \mathbf{A})^{\le\omega} : \forall i \in \mathrm{dom}(\rho)[\ \mathrm{first}(\rho(i)) = S \Rightarrow a \in \mathbf{C}\ ]\}.$$

Two homomorphisms $\pi_L, \pi_R : \mathbf{R} \to \mathbf{A}^{\le\omega}$ are defined as follows: For $a \in \mathbf{A}$ and $c \in \mathbf{C}$,

$$\pi_L(\langle L,a\rangle) = (a), \quad \pi_L(\langle R,a\rangle) = \epsilon, \quad \pi_L(\langle S,c\rangle) = (c),$$
$$\pi_R(\langle L,a\rangle) = \epsilon, \quad \pi_R(\langle R,a\rangle) = (a), \quad \pi_R(\langle S,c\rangle) = (\bar{c}).$$

Then, let $\mathrm{merge}^*(q_1,q_2)$ be the set of elements $\rho \in \mathbf{R}$ satisfying the following conditions:

$$\text{(i)} \quad \mathrm{dom}(\rho) = \begin{cases} \mathrm{dom}(q_1) + \mathrm{dom}(q_2), & \text{if } \mathrm{dom}(q_1), \mathrm{dom}(q_2) \in \omega, \\ \omega, & \text{otherwise,} \end{cases}$$
$$\text{(ii)} \quad \pi_L(\rho) \le_p q_1, \ \pi_R(\rho) \le_p q_2.$$

(2) Another homomorphism $\pi : \mathbf{R} \to \mathbf{A}^{\le\omega}$ is defined as follows: For $a \in \mathbf{A}$ and $c \in \mathbf{C}$,

$$\pi(\langle L,a\rangle) = \pi(\langle R,a\rangle) = (a), \quad \pi(\langle S,c\rangle) = (\tau).$$

Then, let $\mathrm{merge}(q_1,q_2) = \pi[\mathrm{merge}^*(q_1,q_2)]$.

(3) $\qquad \mathrm{merge}_w(q_1,q_2) = \{(q \setminus \tau) : \ q \in \mathrm{merge}(q_1,q_2)\}$.

(4) For $p_1 \subseteq \mathbf{C}^{\le\omega}$, $p_2 \subseteq \mathbf{C}^{\le\omega}$, let $\mathrm{Merge}_w(p_1,p_2) = \bigcup\{\mathrm{merge}_w(q_1,q_2) : \ q_1 \in p_1 \wedge q_1 \in p_1\}$. $\blacksquare$

From the homomorphisms defined above, we have the following lemma:

Lemma 1 *Let* $s_1, s_2, s' \in \mathcal{L}$, $w \in \mathbf{A}^{<\omega}$. *Then*

$$s_1 \| s_2 \xrightarrow{w}_* s' \Leftrightarrow$$
$$\exists\rho \in \mathbf{R}, \exists s_1', s_2' \in \mathcal{L}[\ \pi(\rho) = w \wedge s_1 \xrightarrow{\pi_L(\rho)}_* s_1' \ \wedge \ s_2 \xrightarrow{\pi_R(\rho)}_* s_2' \ \wedge \ s' \equiv s_1' \| s_2'\ ]. \quad \blacksquare$$

Proof. By easy induction on $\mathrm{dom}(w)$. $\blacksquare$

From Merge_w, the semantic operation $\tilde{\|}$ is defined by:

Definition 10 For $p \in \mathbf{D}(\mathcal{C}_{\mathrm{RF}})$, let

$$\mathrm{Trace}_{\mathrm{F}}(p) = \{w \in \mathbf{C}^{<\omega} : \ p[w] \neq \emptyset\},$$
$$\mathrm{Prefix}_{\mathrm{D}}(p) = \{w \in \mathbf{C}^{<\omega} : \ w \cdot (\bot) \in \mathrm{Div}(p)\}.$$

First, four auxiliary operations

$$\mathcal{F}_{\parallel}, \mathcal{R}_{\parallel}, \mathrm{Inf}_{\parallel}, \mathrm{Div}_{\parallel} : (\mathbf{D}(\mathcal{C}_{\mathrm{RF}}) \times \mathbf{D}(\mathcal{C}_{\mathrm{RF}})) \to \mathbf{D}(\mathcal{C}_{\mathrm{RF}})$$

are defined as follows: For $p_1, p_2 \in \mathbf{D}(\mathcal{C}_{\mathrm{RF}})$,

$$\begin{aligned}
\mathcal{F}_{\parallel}(p_1,p_2) = \ & \{w \cdot (\langle \delta, \Gamma \rangle) : \exists w_1 \cdot (\langle \delta, \Gamma_1 \rangle) \in \mathcal{F}^{\circ}(p_1), \exists w_2 \cdot (\langle \delta, \Gamma_1 \rangle) \in \mathcal{F}^{\circ}(p_2) \\
& \quad [\ w \in \mathrm{merge}_w(w_1, w_2) \wedge (\Gamma \subseteq \Gamma_1) \wedge (\Gamma \subseteq \Gamma_2) \\
& \quad \wedge (\mathbf{C} \setminus \Gamma_1) \cap \overline{(\mathbf{C} \setminus \Gamma_2)} = \emptyset \ ]\}, \\
\mathcal{R}_{\parallel}(p_1,p_2) = \ & \{(\langle \delta, \Gamma \rangle) : \exists(\langle \hat{\delta}, \Gamma_1 \rangle) \in \mathcal{R}^{\circ}(p_1), \exists(\langle \hat{\delta}, \Gamma_2 \rangle) \in \mathcal{R}^{\circ}(p_2) \\
& \quad [\ (\Gamma \subseteq \Gamma_1) \wedge (\Gamma \subseteq \Gamma_2) \wedge (\mathbf{C} \setminus \Gamma_1) \cap \overline{(\mathbf{C} \setminus \Gamma_2)} = \emptyset \ ]\}. \\
\mathrm{Inf}_{\parallel}(p_1,p_2) = \ & \mathrm{Merge}_w(\mathrm{Trace}_{\mathrm{F}}(p_1), \mathrm{Inf}^{\circ}(p_2)) \ \cup \ \mathrm{Merge}_w(\mathrm{Trace}_{\mathrm{F}}(p_2), \mathrm{Inf}^{\circ}(p_1)) \\
& \cup (\mathbf{C}^{\omega} \cap \mathrm{Merge}_w(\mathrm{Inf}^{\circ}(p_1), \mathrm{Inf}^{\circ}(p_2))), \\
\mathrm{Div}_{\parallel}(p_1,p_2) = \ & \mathrm{Merge}_w(\mathrm{Trace}_{\mathrm{F}}(p_1), \mathrm{Prefix}_{\mathrm{D}}(p_2)) \cdot (\bot) \\
& \cup \mathrm{Merge}_w(\mathrm{Trace}_{\mathrm{F}}(p_2), \mathrm{Prefix}_{\mathrm{D}}(p_1)) \cdot (\bot) \\
& \cup (\mathbf{C}^{<\omega} \cap \mathrm{Merge}_w(\mathrm{Inf}^{\circ}(p_1), \mathrm{Inf}^{\circ}(p_2))) \cdot (\bot).
\end{aligned}$$

From these, $\tilde{\parallel}$ is defined as follows: For $p_1, p_2 \in \mathbf{D}(\mathcal{C}_{\mathrm{RF}})$,

$$p_1 \mathbin{\tilde{\parallel}} p_2 = \mathcal{F}_{\parallel}(p_1,p_2) \cup \mathcal{R}_{\parallel}(p_1,p_2) \cup \mathrm{Inf}_{\parallel}(p_1,p_2) \cup \mathrm{Div}_{\parallel}(p_1,p_2). \quad \blacksquare$$

From the semantic operations prefix_a ($a \in \mathbf{A}$), $\tilde{+}$, $\tilde{\parallel}$, the compositionality of $\mathcal{C}_{\mathrm{RF}}$ can be established.

Lemma 2 *(Compositionality of $\mathcal{C}_{\mathrm{RF}}$)*
Let $s, s_1, s_2 \in \mathcal{L}$.

(1) *For each $a \in \mathbf{A}$, $\mathcal{C}_{\mathrm{RF}}(a; s) = \mathrm{prefix}_a(\mathcal{C}_{\mathrm{RF}}(s))$.*

(2) $\mathcal{C}_{\mathrm{RF}}(s_1 + s_2) = \mathcal{C}_{\mathrm{RF}}(s_1) \tilde{+} \mathcal{C}_{\mathrm{RF}}(s_2)$.

(3) $\mathcal{C}_{\mathrm{RF}}(s_1 \parallel s_2) = \mathcal{C}_{\mathrm{RF}}(s_1) \mathbin{\tilde{\parallel}} \mathcal{C}_{\mathrm{RF}}(s_2)$. ∎

Proof. By case analysis on the types of elements of $\mathcal{C}_{\mathrm{RF}}(a; s)$, $\mathcal{C}_{\mathrm{RF}}(s_1 + s_2)$, $\mathcal{C}_{\mathrm{RF}}(s_1 \parallel s_2)$, using the definitions of $\mathcal{C}_{\mathrm{RF}}$ and the semantic operations. ∎

6 Correctness of $\mathcal{C}_{\mathrm{RF}}$ with respect to $\mathcal{O}_{\mathrm{WL}}$

The correctness of $\mathcal{C}_{\mathrm{RF}}$ w.r.t. $\mathcal{O}_{\mathrm{WL}}$ is shown by means of an abstraction function $\alpha : \mathbf{D}(\mathcal{C}_{\mathrm{RF}}) \to \mathbf{D}(\mathcal{O}_{\mathrm{WL}})$ defined as follows:

Definition 11 For $p \in \mathbf{D}(\mathcal{C}_{\mathrm{RF}})$, let

$$\alpha(p) = \{w \cdot (\delta) : \exists w, \exists \Gamma[\ w \cdot (\langle \delta, \Gamma \rangle) \in p\ ]\} \ \cup \ \mathrm{Inf}^{\circ}(p) \ \cup \ \mathrm{Div}^{\circ}(p). \quad \blacksquare$$

Note that $\mathcal{R}^{\circ}(p)$ contributes nothing to $\alpha(p)$.

The following proposition follows immediately from the definitions of $\mathcal{O}_{\mathrm{WL}}$, $\mathcal{C}_{\mathrm{RF}}$, and α.

Proposition 1 *For every $s \in \mathcal{L}$, $\mathcal{O}_{\mathrm{WL}}(s) = \alpha(\mathcal{C}_{\mathrm{RF}}(s))$.* ∎

By this and the compositionality of $\mathcal{C}_{\mathrm{RF}}$, the *correctness* of $\mathcal{C}_{\mathrm{RF}}$ for $\hat{\mathcal{L}}$ w.r.t. $\mathcal{O}_{\mathrm{WL}}$ can be established (for a proof for $\mathcal{L}$ see the Appendix).

Lemma 3 *(Correctness of $\mathcal{C}_{\mathrm{RF}}$ for $\hat{\mathcal{L}}$)*
Let $s_1, s_2 \in \mathcal{L}$. If $\mathcal{C}_{\mathrm{RF}}(s_1) = \mathcal{C}_{\mathrm{RF}}(s_2)$, then the following holds for every $S \in \hat{\mathcal{L}}[X]$:

$$\mathcal{O}_{\mathrm{WL}}(S[s_1/X]) = \mathcal{O}_{\mathrm{WL}}(S[s_2/X]). \quad \blacksquare \tag{10}$$

Proof. This follows straightforwardly from Lemma 2 and Proposition 1. Formally, this can be established by induction on $\deg(S)$, the number of operators included in S and in the scope of no μ-statement.
Induction Base: Suppose $\deg(S) = 0$. Then either $S \equiv D$ or $S \equiv X$ or $X \equiv \mu Y(S')$ with some $Y \in \mathcal{V}$, $S' \in \mathcal{L}$. In the last case, $X \notin \mathrm{FV}(S')$ by the condition (3) in Definition 2. Thus in all cases, $S[s_1/X] \equiv S[s_2/X]$, and therefore, (10) holds.
Induction Step: Assume that (10) holds for every $S' \in \mathcal{L}$ with $\deg(S') \leq n$. Let $S \in \mathcal{L}$ with $\deg(S) = n+1$. Then either $S \equiv a; S_1$ or $S \equiv S_1 + S_2$ or $S \equiv S_1 \parallel S_2$, with some $S_i \in \mathcal{L}$ such that $\deg(S_i) \leq n$ ($i = 1, 2$). In all cases, (10) follows from the induction hypothesis and Lemma 2. ∎

7 Full Abstractness of $\mathcal{C}_{\mathrm{RF}}$ with respect to $\mathcal{O}_{\mathrm{WL}}$

The full abstractness of $\mathcal{C}_{\mathrm{RF}}$ w.r.t. $\mathcal{O}_{\mathrm{WL}}$ can be established under the assumption that

the communication set $\mathbf{C}$ is infinite.[2] (11)

As a preliminary, for $s \in \mathcal{L}$, let $\mathcal{A}(s)$, the *alphabet* of s, be defined by:

Definition 12 $\mathcal{A}(s) = \{c \in \mathbf{C} : \exists w, \exists s'[\ s \overset{w}{\Longrightarrow} s' \wedge c \in \mathrm{act}(s')\ ]\}$. ∎

The following proposition follows from the definition of $\mathcal{L}$.

Proposition 2 $\forall s \in \mathcal{L}[\ \mathcal{A}(s)$ *is finite* $]$. ∎

Theorem 1 *(Full Abstractness of $\mathcal{C}_{\mathrm{RF}}$)*
For $s_1, s_2 \in \mathcal{L}$,

$$\begin{array}{l} \mathcal{C}_{\mathrm{RF}}(s_1) = \mathcal{C}_{\mathrm{RF}}(s_2) \\ \Leftrightarrow \forall S \in \mathcal{L}[X][\ \mathcal{O}_{\mathrm{WL}}(S[s_1/X]) = \mathcal{O}_{\mathrm{WL}}(S[s_2/X])\ ]. \end{array} \blacksquare$$

Proof. The $\Rightarrow$-part is the statement of Lemma 3. For establishing the $\Leftarrow$-part, it suffices to show

$$\mathcal{C}_{\mathrm{RF}}(s_1) \neq \mathcal{C}_{\mathrm{RF}}(s_2) \Rightarrow \exists S \in \mathcal{L}[X][\ \mathcal{O}_{\mathrm{WL}}(S[s_1/X]) \neq \mathcal{O}_{\mathrm{WL}}(S[s_2/X])\ ].$$

Let $s_1, s_2 \in \mathcal{L}$, and suppose $\mathcal{C}_{\mathrm{RF}}(s_1) \neq \mathcal{C}_{\mathrm{RF}}(s_2)$. When

$$\mathrm{Inf}(s_1) \neq \mathrm{Inf}(s_2) \text{ or } \mathrm{Div}(s_1) \neq \mathrm{Div}(s_2),$$

it follows immediately that $\mathcal{O}_{\mathrm{WL}}(s_1) \neq \mathcal{O}_{\mathrm{WL}}(s_2)$.
Otherwise, there are two cases.
Case 1. Suppose $\mathcal{F}(s_1) \neq \mathcal{F}(s_2)$. Then one can construct an appropriate statement T called a *tester* such that $\mathcal{O}_{\mathrm{WL}}(s_1 \parallel T) \neq \mathcal{O}_{\mathrm{WL}}(s_2 \parallel T)$.[3]
One has either

$$\textbf{(i)}\ \exists w \cdot (\langle \delta, \Gamma \rangle) \in \mathcal{F}(s_1) \setminus \mathcal{F}(s_2) \quad \text{or} \quad \textbf{(ii)}\ \exists w \cdot (\langle \delta, \Gamma \rangle) \in \mathcal{F}(s_2) \setminus \mathcal{F}(s_1). \tag{12}$$

The former case is considered. We can assume that Γ is finite by Proposition 2. Let $\Gamma = \{c_1, \cdots, c_n\}$, and let us take

$$c \in (\mathbf{C} \setminus (\mathcal{A}(s_1) \cup \mathcal{A}(s_2) \cup \overline{\mathcal{A}(s_2)})). \tag{13}$$

The set $(\mathbf{C} \setminus (\mathcal{A}(s_1) \cup \mathcal{A}(s_2) \cup \overline{\mathcal{A}(s_2)})$ is non-empty by Proposition 2 under the assumption (11). Setting $\Omega \equiv \mu X(\tau; X)$, $T' \equiv D + (\overline{c_1}; \Omega) + \cdots + (\overline{c_n}; \Omega)$, and $T \equiv c; T'$, it follows immediately from (12) (i) and the definition of T that

$$w \cdot (c) \cdot (\delta) \in \mathcal{O}_{\mathrm{WL}}(s_1 \parallel T). \tag{14}$$

Let us show, by contradiction, that

$$w \cdot (c) \cdot (\delta) \notin \mathcal{O}_{\mathrm{WL}}(s_2 \parallel T). \tag{15}$$

Assume that this does not hold. Then, by the definition of $\mathcal{O}_{\mathrm{WL}}$, there are $s', s'' \in \mathcal{L}$, $\tilde{w} \in \mathbf{A}^{<\omega}$, and $k \geq 0$ such that

$$s_2 \parallel T \xrightarrow{\tilde{w}} s' \xrightarrow{(c)\cdot\tau^k} s'' \wedge (\tilde{w} \setminus \tau) = w \wedge \mathrm{act}(s'') \subseteq \mathbf{C}.$$

By (13), the action c must stem from T. Moreover, by (13), there can be no synchronization between s_2 and T before T has performed the action c. Thus the actions in $\tilde{w}$ must stem from s_2, and therefore, there exists s_2' such that

$$\textbf{(i)}\ s_2 \overset{\tilde{w}}{\Longrightarrow} s_2' \wedge s' \equiv s_2' \parallel T, \quad \textbf{(ii)}\ (s_2' \parallel T) \xrightarrow{c} (s_2' \parallel T') \xrightarrow{\tau^k} s'', \quad \textbf{(iii)}\ \mathrm{act}(s'') \subseteq \mathbf{C}. \tag{16}$$

By (16) (ii) and Lemma 1, there are $\rho \in \mathbf{R}$, s_2'', and T'' such that

$$\pi(\rho) = \tau^k \wedge s_2' \xrightarrow{\pi_L(\rho)}_* s_2'' \wedge T' \xrightarrow{\pi_R(\rho)}_* T'' \wedge s'' \equiv s_2'' \parallel T''. \tag{17}$$

Let us show, by contradiction, that

$$\neg \exists i \in \mathrm{dom}(\rho)[\ \mathrm{first}(\rho(i)) \in \{R, S\}\ ]. \tag{18}$$

[2]This assumption might seem too strong when we consider hardware systems where *communications* are regarded as physical ports. However, for software systems where *communications* are regarded as identifiers (such as *entry identifiers* of Ada), this assumption seems reasonable. A similar assumption is given by Milner for characterizing *observation congruence* (cf. [Mil 89] § 7.2).

[3]The variable T is used to denote a statement when it is considered a tester, while the typical variable for the set of statements is s.

If this does not hold, then one has, by the form of T', that $T'' \equiv \Omega$, and therefore, $\mathrm{act}(s'') = \mathrm{act}(s_2'' \parallel \Omega) \ni \tau$, which contradicts (16) (iii). Hence one has (18).

Thus one has $\forall i \in \mathrm{dom}(\rho)[\ \mathrm{first}(\rho(i)) = L\ ]$, and therefore, $\tau^k = \pi(\rho) = \pi_L(\rho)$, $s_2' \xrightarrow{\tau^k} s_2''$, and $s'' \equiv s_2'' \parallel T'$. By this, (16) (iii), and (17), one has $\mathrm{act}(s_2'') \subseteq \mathbf{C}$ and $\mathrm{act}(s_2'') \cap \mathrm{act}(T') = \emptyset$. Thus, by (16) (i), one has $w \cdot (\langle \delta, \Gamma \rangle) \in \mathcal{F}(s_2)$, which contradicts (12) (i).

Thus one has (15); it follows from (14) and (15) that $\mathcal{O}_{\mathrm{WL}}(s_1 \parallel T) \neq \mathcal{O}_{\mathrm{WL}}(s_2 \parallel T)$.

Case 2. Suppose $\mathcal{F}(s_1) = \mathcal{F}(s_2)$ and $\mathcal{R}(s_1) \neq \mathcal{R}(s_2)$. Then either

$$\textbf{(i)}\ \exists(\langle \hat{\delta}, \Gamma \rangle) \in \mathcal{R}(s_1) \setminus \mathcal{R}(s_2) \quad \text{or} \quad \textbf{(ii)}\ \exists(\langle \hat{\delta}, \Gamma \rangle) \in \mathcal{R}(s_2) \setminus \mathcal{R}(s_1). \tag{19}$$

The former case is considered. Since $(\langle \tilde{\delta}, \Gamma \rangle) \in \mathcal{R}(s_1)$, one has

$$\mathrm{act}(s_1) \subseteq \mathbf{C}. \tag{20}$$

Moreover, by (19) (i), one has $(\langle \delta, \Gamma \rangle) \in \mathcal{F}(s_1)$, and therefore, $(\langle \delta, \Gamma \rangle) \in \mathcal{F}(s_2)$.

Thus there exits s_2' such that

$$s_2 \stackrel{\epsilon}{\Longrightarrow} s_2' \wedge (\mathrm{act}(s_2') \subseteq \mathbf{C}) \wedge (\Gamma \cap \mathrm{act}(s_2') = \emptyset). \tag{21}$$

Since $(\langle \hat{\delta}, \Gamma \rangle) \notin \mathcal{R}(s_2)$ by (19) (i), s_2' cannot be s_2 itself; thus there exists s_2'' such that

$$s_2 \xrightarrow{\tau} s_2'' \stackrel{\epsilon}{\Longrightarrow} s_2' \tag{22}$$

Thus $\tau \in \mathrm{act}(s_2)$. Let us take a context $S \equiv X + (\tau; \Omega)$.

First, one has

$$(\delta) \notin \mathcal{O}_{\mathrm{WL}}(S[s_1/X]) \equiv \mathcal{O}_{\mathrm{WL}}(s_1 + (\tau; \Omega)), \tag{23}$$

since by (20), there is no s_1' such that $(s_1 + (\tau; \Omega)) \stackrel{\epsilon}{\Longrightarrow} s_1'$ and $(\mathrm{act}(s_1') \subseteq \mathbf{C})$.

Next, one has

$$(\delta) \in \mathcal{O}_{\mathrm{WL}}(S[s_2/X]) \equiv \mathcal{O}_{\mathrm{WL}}(s_1 + (\tau; \Omega)), \tag{24}$$

since it follows from (21) and (22), that $(s_2 + (\tau; \Omega)) \stackrel{\epsilon}{\Longrightarrow} s_2'$ and $(\mathrm{act}(s_2') \subseteq \mathbf{C})$.

By (23) and (24), one has $\mathcal{O}_{\mathrm{WL}}(S[s_1/X]) \neq \mathcal{O}_{\mathrm{WL}}(S[s_2/X])$. ■

8 Concluding Remarks

We conclude this paper with remarks about possible extensions of the reported result. There are two directions for such extensions.

One is to investigate the same full abstractness problem for other languages that are extensions of $\mathcal{L}$. The set of operators of $\mathcal{L}$ is rather restricted. The operators *sequential composition* and *abstraction* of ACP_τ (cf. [BK 85]), as well as *restriction* and *relabeling* of CCS (cf. [Mil 80]) are good candidates to add to $\mathcal{L}$. The author conjectures that $\mathcal{C}_{\mathrm{RF}}$ is a congruence w.r.t. any set of operators defined on the basis of a *transition system specification* in the so-called *SOS* format (cf. [BIM 88]). A similar full abstractness problem for *nonuniform languages* such as the ones treated in [HBR 90], also remains for future study.

The other direction is to investigate denotational construction of $\mathcal{C}_{\mathrm{RF}}$ in the order-theoretic or metric topological setting. It is hoped to accomplish this by means of the construction method in [BHR 84] or the one in [Rut 89], with some modification if necessary. However, neither of them can be used as it is, as described below.

There are two difficulties in using the standard ordering in [BHR 84], i.e., *inverse inclusion*. First, the operation $\tilde{\parallel}$ is not in general continuous w.r.t. this ordering. For example, let
$p_n = \bigcup\{\mathcal{C}_{\mathrm{RF}}(c^k; D) : k \geq n\}$ $(n \geq 1)$, and $p' = \bigcup\{\mathcal{C}_{\mathrm{RF}}(c^k; D) : k \geq 1\}$.
Then $\forall n \geq [\ (\langle \delta, \{c, \bar{c}\} \rangle) \in p_n \,\tilde{\parallel}\, p'\ ]$, but $(\langle \delta, \{c, \bar{c}\} \rangle) \notin \bigcap_{n \geq 1}(p_n) \,\tilde{\parallel}\, p'$. Second, for some recursively defined statement $\mu X(S)$, the least upper bound of the iteration sequence generated by the interpretation of S, does not coincide with the intended meaning. For example, we would like to define $\mathcal{C}_{\mathrm{RF}}$ so that $\mathcal{C}_{\mathrm{RF}}(\mu X(\tau; X)) = \{(\perp)\}$. However the iteration sequence generated by the function $(\lambda p \in \mathbf{P} : \mathrm{Prefix}_\tau(p))$ with the initial point $\mathrm{CHAOS} = \mathbf{B}(\mathcal{C}_{\mathrm{RF}})$ gives a rather different value $\mathbf{B}(\mathcal{C}_{\mathrm{RF}}) \setminus \{\langle \tilde{\delta}, \Gamma \rangle : \Gamma \subseteq \mathbf{C}\}$.

As Rutten did in [Rut 89], one can define a distance d on $\mathbf{B}(\mathcal{C}_{\mathrm{RF}})$ by means of *truncation*; this distance induce the so-called *Hausdorff metric* on $\wp_{\mathrm{cls}}(\mathbf{B}(\mathcal{C}_{\mathrm{RF}}))$, the domain of *closed subsets* of $\mathbf{B}(\mathcal{C}_{\mathrm{RF}})$. There are two difficulties in using this metric: First, it is not known whether $\mathcal{C}_{\mathrm{RF}}(s)$ is closed in $\mathbf{D}(\mathcal{C}_{\mathrm{RF}})$ or not, for $s \in \mathcal{L}$. Second, unlike in the *strong* semantics of [Rut 89], the operation $\tilde{\parallel}$ is not in general *non-expansive*, even if $\mathcal{C}_{\mathrm{RF}}(s)$ is closed for every $s \in \mathcal{L}$. For example, let $s_1 \equiv (c; D)$, $s_1' \equiv (c; \Omega)$, $s_2 \equiv (\bar{c}; D)$, and $s_2' \equiv (\bar{c}; \Omega)$, where Ω is the statement defined in the proof of Theorem 1. Then $d(\mathcal{C}_{\mathrm{RF}}(s_1), \mathcal{C}_{\mathrm{RF}}(s_1')) = (1/2)$, $d(\mathcal{C}_{\mathrm{RF}}(s_2), \mathcal{C}_{\mathrm{RF}}(s_2')) = (1/2)$, but $d(\mathcal{C}_{\mathrm{RF}}(s_1 \parallel s_2), \mathcal{C}_{\mathrm{RF}}(s_1' \parallel s_2')) = 1$.

Acknowledgements

The author would like to thank Prof. J. de Bakker who introduced him to the full abstractness problem for concurrency and encouraged him to tackle it, and Dr. J. Rutten for his helpful comments. He would also like to thank his colleagues at NTT Software Laboratories, especially Mr. K. Yamanaka for his valuable comments which led to Remark 1 (2), Mr. A. Takura for his useful comments which let to Example 1, and Dr. H. Ichikawa for his helpful comments and encouragement.

References

[AP 86] K. APT, G. PLOTKIN (1986). Countable nondeterminism and random assignment, *Journal of the Association for Computer Machinery, Vol.33, No.4*, pp. 724-767.

[BZ 82] J.W. DE BAKKER, J.I. ZUCKER (1982). Processes and the denotational semantics of concurrency, *Inform. and Control* **54**, pp 70–120.

[BK 85] J.A. BERGSTRA, J.W. KLOP (1985). Algebra of communicating processes with abstraction, *Theoretical Computer Science, Vol.37, No.1*, pp. 77–121.

[BKO 88] J.A. BERGSTRA, J.W. KLOP, E.-R. OLDEROG (1988). Readies and failures in the algebra of communicating processes, *SIAM J. of Computing Vol.17, No.6*, pp. 1134–1177.

[BIM 88] B. BLOOM, S. ISTRAIL, A.R. MEYER (1988). Bisimulation can't be traced: preliminary report, in *Proceedings of the 15th POPL*, San Diego, California, 1988, pp.229–239.

[Bro 83] S.D. BROOKES (1983). On the relationship of CCS and CSP, in *Proceedings 10th ICALP, Barcelona, Lecture Notes in Computer Science, Vol.154*, pp. 83–96, Springer, Berlin.

[BHR 84] S.D. BROOKES, C.A.R. HOARE, A.W. ROSCOE (1984). A theory of communicating sequential processes, *Journal of the Association for Computer Machinery, Vol.31*, pp. 560–599.

[BR 84] S.D. BROOKES, A.W. ROSCOE (1984). An improved failures model for communicating processes, Lecture Notes in Computer Science, Vol.197, pp. 281–305.

[HP 79] M. HENNESSY, G.D. PLOTKIN (1979). Full abstraction for a simple parallel programming language, in *Proceedings 8th MFCS* (J. BEČVÁŘ ed.), *Lecture Notes in Computer Science, Vol. 74*, pp. 108–120, Springer.

[HBR 90] E. HORITA, J.W. DE BAKKER, J.J.M.M. RUTTEN (1990). *Fully abstract denotational models for nonuniform concurrent languages, CWI Report CS-R9027*, Amsterdam.

[Kun 80] K. KUNEN (1981). *Set theory*, North-Holland, Amsterdam.

[Mil 73] R. MILNER (1973). Processes: a mathematical model of computing agents, in *Proceedings of Logic Colloquium 73* (H.E. ROSE, J.C. SHEPHERDSON, eds.), pp. 157–173, North-Holland.

[Mil 80] R. MILNER (1980). *A Calculus of communicating systems, Lecture Notes in Computer Science Vol.92*, Springer.

[Mil 83] R. MILNER (1983). Calculi for synchrony and asynchrony, *Theoretical Computer Science, Vol.25*, pp. 267–310.

[Mil 85] R. MILNER (1985). Lectures on a calculus for communicating systems, in *Seminar on Concurrency* (S.D. BROOKES, A.W. ROSCOE, AND G. WINSKEL, eds.), *Lecture Notes in Computer Science Vol.197*, pp. 197–220, Springer.

[Mil 89] R. MILNER (1989), *Communication and concurrency*, Prentice Hall International.

[Plo 81] G.D. PLOTKIN (1981). *A structured approach to operational semantics*, Report DAIMI FN-19, Computer Science Department, Aarhus University.

[Rut 89] J.J.M.M. RUTTEN (1989). Correctness and full abstraction of metric semantics for concurrency, in *Linear Time, Branching Time and Partial Order in Logics and Models for Concurrency* (J.W. DE BAKKER, W.P. DE ROEVER, G. ROZENBERG, eds.), *Lecture Notes in Computer Science Vol.354*, pp. 628–658, Springer.

Appendix

A Proof of Correctness of $\mathcal{C}_{\text{RF}}$ for the Full Language $\mathcal{L}$

A semantic model $\mathcal{M} : \mathcal{L} \to \mathbf{D}(\mathcal{M})$ is called a *congruence* for $\mathcal{L}$ iff it satisfies the following:

$$\forall s_1, s_2 \in \mathcal{L}[\ \mathcal{M}(s_1) = \mathcal{M}(s_2) \Leftrightarrow \forall S \in \mathcal{L}[X][\ \mathcal{M}(S[s_1/X]) = \mathcal{M}(S[s_2/X])\ ]].$$

Since $\mathcal{C}_{\text{RF}}$ respects $\mathcal{O}_{\text{WL}}$ (cf. Proposition 1), the correctness of $\mathcal{C}_{\text{RF}}$ w.r.t. $\mathcal{O}_{\text{WL}}$ follows immediately from the proposition that $\mathcal{C}_{\text{RF}}$ is a congruence for $\mathcal{L}$. Therefore, let us prove that $\mathcal{C}_{\text{RF}}$ is a congruence for $\mathcal{L}$.

Let $s_1, s_2 \in \mathcal{L}$ such that $\mathcal{C}_{\text{RF}}(s_1) = \mathcal{C}_{\text{RF}}(s_2)$, and $S \in \mathcal{L}[X]$. Let us prove the following:

$$\mathcal{C}_{\text{RF}}(S[s_1/X]) = \mathcal{C}_{\text{RF}}(S[s_2/X]). \tag{25}$$

We will prove this, by induction on the *length of inferences* through which transitions $s \xrightarrow{a} s'$ are proved, which is analogous to Milner's proof of the fact that the so-called *strong equivalence* is a congruence (cf. [Mil 83], § 4).

Some notational preliminaries are needed for the proof.

Notation 4 For $s, s' \in \mathcal{L}$, $a \in \mathbf{A}$, and $n \in \omega$, let us write $\vdash_{(n)} s \xrightarrow{a} s'$, to denote that there is an inference with length n through which $s \xrightarrow{a} s'$ is proved.

We write $\vdash_{(\leq n)} s \xrightarrow{a} s'$, to denote that $\exists k \leq n[\ \vdash_{(k)} s \xrightarrow{a} s'\ ]$. ∎

One obtain immediately, by the definition of the transition relation $\to$, that

$$s \xrightarrow{a} s' \Leftrightarrow \exists n \in \omega[\ \vdash_{(n)} s \xrightarrow{a} s'\ ].$$

In order to prove (25), it suffices to prove the following:

$$\begin{cases} \text{(i)} & \mathcal{R}(S[s_1/X]) = \mathcal{R}(S[s_2/X]), \\ \text{(ii)} & \mathcal{F}(S[s_1/X]) = \mathcal{F}(S[s_2/X]), \\ \text{(iii)} & \text{Inf}(S[s_1/X]) = \text{Inf}(S[s_2/X]), \\ \text{(iv)} & \text{Div}(S[s_1/X]) = \text{Div}(S[s_2/X]). \end{cases} \tag{26}$$

The propositions (i), (ii), (iii), and (iv) of (26) will be proved in this order.

Lemma 4 *Let $s_1, s_2 \in \mathcal{L}$ such that $\mathcal{C}_{\text{RF}}(s_1) = \mathcal{C}_{\text{RF}}(s_2)$, and $S \in \mathcal{L}[X]$. Then, (26) (i) holds, i.e., $\mathcal{R}(S[s_1/X]) = \mathcal{R}(S[s_2/X])$.* ∎

Proof. In order to prove $\mathcal{R}(S[s_1/X]) = \mathcal{R}(S[s_2/X])$, it suffices, by the definition of $\mathcal{R}$ (cf. Definition 6), to prove the following:

$$\begin{cases} \text{(i)} & \forall S \in \mathcal{L}[X][\ \exists s_1'[\ S[s_1/X] \xrightarrow{\tau} s_1'\ ] \Rightarrow \exists s_2'[\ S[s_2/X] \xrightarrow{\tau} s_2'\ ]], \\ \text{(ii)} & \forall S \in \mathcal{L}[X][\ \neg\exists s_2'[\ S[s_2/X] \xrightarrow{\tau} s_2'\ ] \Rightarrow \\ & \qquad \forall c[\ \exists s_1'[\ S[s_1/X] \xrightarrow{c} s_1'\ ] \Rightarrow \exists s_2'[\ S[s_2/X] \xrightarrow{c} s_2'\ ]]]. \end{cases} \tag{27}$$

For $n \in \omega$, let

$$\Phi'_{\mathcal{R}}(n) \Leftrightarrow \forall S \in \mathcal{L}[X][\ \exists s_1'[\ \vdash_{(n)} S[s_1/X] \xrightarrow{\tau} s_1'\ ] \Rightarrow \exists s_2'[\ S[s_2/X] \xrightarrow{\tau} s_2'\ ]],$$
$$\Phi''_{\mathcal{R}}(n) \Leftrightarrow \forall S \in \mathcal{L}[X][\ \neg\exists s_2'[\ S[s_2/X] \xrightarrow{\tau} s_2'\ ] \Rightarrow$$
$$\forall c[\ \exists s_1'[\vdash_{(n)} S[s_1/X] \xrightarrow{c} s_1'\ ] \Rightarrow \exists s_2'[\ S[s_2/X] \xrightarrow{c} s_2'\ ]]].$$

We will prove, by induction, that $\forall n[\ \Phi'_{\mathcal{R}}(n) \wedge \Phi''_{\mathcal{R}}(n)\ ]$ holds.

Induction Base: First, we will prove $\Phi'_{\mathcal{R}}(0)$. Suppose $\exists s_1'[\ \vdash_{(0)} S[s_1/X] \xrightarrow{\tau} s_1'\ ]$. We distinguish two possible cases according to the form of S.

Case 1: Suppose $S \equiv X$. Then one has,

$S[s_1/X] \equiv s_1 \xrightarrow{\tau} s_1'$, and therefore, $\exists s_2'[\ S[s_2/X] \equiv s_2 \xrightarrow{\tau} s_2'\ ]$,

since $\mathcal{C}_{\text{RF}}(s_1) = \mathcal{C}_{\text{RF}}(s_2)$.

Case 2: Otherwise, $S \equiv (\tau; S')$ for some S', and therefore, $S[s_2/X] \equiv (\tau; S'[s_2/X])$. Thus, $S[s_2/X] \xrightarrow{\tau} S'[s_2/X]$, and therefore, $\exists s_2'[\ S[s_2/X] \xrightarrow{\tau} s_2'\ ]$.

The other part $\Phi''_{\mathcal{R}}(0)$ can be established by a similar case analysis to the above one. Thus one has $\Phi'_{\mathcal{R}}(0) \wedge \Phi'_{\mathcal{R}}(0)$.

Induction Step: For $k \in \omega$, assume $\forall i \leq k[\ \Phi'_{\mathcal{R}}(i) \wedge \Phi''_{\mathcal{R}}(i)\ ]$. Let us prove $\Phi'_{\mathcal{R}}(k+1) \wedge \Phi''_{\mathcal{R}}(k+1)$.

We will prove only $\Phi'_{\mathcal{R}}(k+1)$; the other part $\Phi''_{\mathcal{R}}(k+1)$ is proved similarly. Let $S \in \mathcal{L}[X]$, and suppose

$$\exists s_1'[\ \vdash_{(k+1)} S[s_1/X] \xrightarrow{\tau} s_1'\ ]. \tag{28}$$

Let us show

$$\exists s_2'[\ S[s_2/X] \xrightarrow{\tau} s_2'\ ]. \tag{29}$$

We distinguish 5 cases according to the form S, i.e., one of the following holds: $S \equiv X$, $S \equiv (\tau; S')$, $S \equiv (S' + S'')$, $S \equiv (S' \parallel S'')$, or $S \equiv \mu Y(S')$.

If $S \equiv X$, then (29) is obtained immediately.

Out of the other 4 cases, we consider the case where $S \equiv \mu Y(S')$; in the other 3 cases the same result is obtained similarly. If $Y \equiv X$, then (29) is obtained immediately. Suppose $Y \not\equiv X$. Then, by (28), one has

$$\vdash_{(k+1)} \mu Y(S')[s_1/X] \equiv \mu Y(S'[s_1/X]) \xrightarrow{\tau} s_1'.$$

By this and the definition of $\longrightarrow$, one has

$$\vdash_{(k)} S'[s_1/X][\mu Y(S'[s_1/X])/Y] \equiv S'[\mu Y(S')/Y][s_1/X] \xrightarrow{\tau} s_1'.$$

By this and the induction hypothesis, one has

$$\exists s_2'[\ S'[\mu Y(S')/Y][s_2/X] \equiv S'[s_2/X][\mu Y(S'[s_2/X])/Y] \xrightarrow{\tau} s_2'\ ].$$

Thus, by the recursion rule, one has

$$\mu Y(S'[s_2/X]) \equiv S[s_2/X] \xrightarrow{\tau} s_2'. \quad \blacksquare$$

The following notations are introduced as a preliminary to the proof of (26) (ii).

Notation 5 For $w_1, w_2 \in \mathbf{A}^{\leq\omega}$, let $w_1 \approx^+ w_2$ iff $(w_1 \setminus \tau) = (w_2 \setminus \tau)$ and $(w_1 \neq \epsilon \Leftrightarrow w_2 \neq \epsilon)$. $\blacksquare$

Lemma 5 *Let $s_1, s_2 \in \mathcal{L}$ such that $\mathcal{C}_{\mathrm{RF}}(s_1) = \mathcal{C}_{\mathrm{RF}}(s_2)$, and $S \in \mathcal{L}[X]$. Then, (26) (ii) holds, i.e., $\mathcal{F}(S[s_1/X]) = \mathcal{F}(S[s_2/X])$.* $\blacksquare$

Proof. In order to prove $\mathcal{F}(S[s_1/X]) = \mathcal{F}(S[s_2/X])$, it suffices to prove that the proposition $\Phi_{\mathcal{F}}(n,k)$ defined below, holds for every $n, k \in \omega$:

$$\begin{aligned}
&\Phi_{\mathcal{F}}(n,k) \Leftrightarrow \\
&\forall a \in \mathbf{A}, \forall \tilde{w}_1 \in \mathbf{A}^n, \forall X, \forall S[\ \exists s_1'', s_1'[\ \vdash_{(k)} S[s_1/X] \xrightarrow{a} s_1'' \ \wedge\ s_1'' \xrightarrow{\tilde{w}_1} s_1' \\
&\qquad\qquad \wedge\ \mathrm{act}(s_1) \subseteq \mathbf{C} \ \wedge\ \mathrm{act}(s_1) \cap X = \emptyset\ ] \\
&\qquad\qquad \Rightarrow \exists w_2, \exists s_2'[\ w_2 \approx^+ (a) \cdot \tilde{w}_1 \ \wedge\ S[s_2/X] \xrightarrow{w_2} s_2' \\
&\qquad\qquad \wedge\ \mathrm{act}(s_2') \subseteq \mathbf{C} \ \wedge\ \mathrm{act}(s_2') \cap \mathbf{C} = \emptyset\ ]].
\end{aligned} \tag{30}$$

This is achieved by iterated induction. That is, the claim that $\forall n[\ \forall k[\ \Phi_{\mathcal{F}}(n,k)\ ]]$ is proved by induction on n, where for each n, the claim that $\forall k[\ \Phi_{\mathcal{F}}(n,k)\ ]$ is proved by induction on k.

Note that $\Phi_{\mathcal{F}}(0,0)$ is proved by means of Lemma 4. The induction steps are similar to the proof of Lemma 4. $\blacksquare$

For the proof of (26) (iii), (iv), the notion of *strong bisimulation* is assumed to be known (cf., e.g., [Mil 83], § 4), and the following notations and definition are introduced:

Notation 6 Let $s, s_1, s_2 \in \mathcal{L}$, $a \in \mathbf{A}$, and $w \in \mathbf{A}^{\leq\omega}$.

(1) The strong bisimulation is denoted by $\sim$. That is, $s_1 \sim s_2$ iff s_1 and s_2 are bisimilar.

(2) $s_1 \xrightarrow{a}_b s_2$ iff $\exists s_2'[\ s_1 \xrightarrow{a} s_2' \wedge s_2' \sim s_2\ ]$.

(3) $s_1 \xrightarrow{w}_{*b} s_2$ iff $\exists s_2'[\ s_1 \xrightarrow{w}_* s_2' \wedge s_2' \sim s_2\ ]$.

(4) When $w \in \mathbf{A}^\omega$, we write $s \xrightarrow{w}_*$ to denote that $\exists s'[s \xrightarrow{w}_* s']$. When $w = \mathbf{A}^{<\omega}$, we write $s \xrightarrow{w}_*$ to denote that $\exists (s_i)_{i\in\omega}[\ s = s_0 \wedge \forall i \in \omega[\ s_i \xrightarrow{w(i)} s_{i+1}\ ]]$. $\blacksquare$

Definition 13 For $w \in \mathbf{A}^{\leq\omega}$, let

$$\theta_\perp(w) = \begin{cases} (w \setminus \tau) \cdot (\perp) & \text{if } \exists \tilde{w}[\ w = \tilde{w} \cdot \tau^\omega\ ], \\ (w \setminus \tau) & \text{otherwise.} \end{cases} \quad \blacksquare$$

The following lemma essentially used in the proof of (26) (iii), (iv).

Lemma 6 *Let $s_1, s_2 \in \mathcal{L}$ such that $\mathcal{C}_{\mathrm{RF}}(s_1) = \mathcal{C}_{\mathrm{RF}}(s_2)$. Then for $w \in \mathbf{A}^+$, $w' \in \mathbf{A}^{\leq\omega}$, and $S \in \mathcal{L}[X]$, the following holds:*

$$
\begin{array}{l}
(w \setminus \tau) \neq \epsilon \wedge S[s_1/X] \xrightarrow{w \cdot w'}_{*} \\
\quad \Rightarrow \exists \tilde{w} \in \mathbf{A}^{+}, \exists w'' \in \mathbf{A}^{\leq \omega}, \exists \tilde{s} \in \mathcal{L}, \exists \tilde{S} \in \mathcal{L}[X] \\
\qquad [\ (w \setminus \tau) = (\tilde{w} \setminus \tau) \wedge \theta_{\perp}(w') = \theta_{\perp}(w'') \wedge \tilde{s} \parallel \tilde{S}[s_1/X] \xrightarrow{w''}_{*} \\
\qquad \wedge\ S[s_2/X] \xrightarrow{\tilde{w}}_{*b} \tilde{s} \parallel \tilde{S}[s_2/X]\ ]. \quad \blacksquare
\end{array}
$$

Proof. This lemma is proved in a similar fashion to the proof of Lemma 5, by iterated induction on $\mathrm{dom}(w)$ and the number k such that $\exists s_1'[\ \vdash_{(k)} S[s_1/X] \xrightarrow{w(0)} s_1' \wedge s_1' \xrightarrow{\mathrm{rest}(w) \cdot w'}_{*}\ ]$.

For $\mathrm{dom}(w) = 0$ and $k = 0$, we take $\tilde{s}$ and $\tilde{S}$ as follows: If $S = X$, then we take $\tilde{s}$ such that $S[s_2/X] \equiv s_2 \overset{(w \setminus \tau)}{\Longrightarrow} \tilde{s} \xrightarrow{w'}$ and $\tilde{S} = D$. Otherwise, $S \equiv (w(0); S')$ for some S', and we take $\tilde{s} \equiv D$ and $\tilde{S} \equiv S'$. $\blacksquare$

Let us prove (26) (iii) by means of Lemma 6.

Lemma 7 *Let $s_1, s_2 \in \mathcal{L}$ such that $\mathcal{C}_{\mathrm{RF}}(s_1) = \mathcal{C}_{\mathrm{RF}}(s_2)$, and $S \in \mathcal{L}[X]$. Then, (26) (iii) holds, i.e.,* $\mathrm{Inf}(S[s_1/X]) = \mathrm{Inf}(S[s_2/X])$. $\blacksquare$

Proof. In order to prove $\mathrm{Inf}(S[s_1/X]) = \mathrm{Inf}(S[s_2/X])$, it suffices to prove $\mathrm{Inf}(S[s_1/X]) \subseteq \mathrm{Inf}(S[s_2/X])$ and $\mathrm{Inf}(S[s_2/X]) \subseteq \mathrm{Inf}(S[s_1/X])$.
We will prove the first part. In order to show this, it suffices to prove the following:

$$
\begin{array}{l}
\forall (c_n)_{n \in \omega} \in \mathbf{C}^{\omega}, \forall w \in \mathbf{A}^{\omega} \\
[\ (w \setminus \tau) = (c_n)_{n \in \omega} \wedge S[s_1/X] \xrightarrow{w}_{*} \Rightarrow \\
\ \exists \tilde{s}_0 \in \mathcal{L}, \exists \tilde{S}_0 \in \mathcal{L}[X], \exists (\langle \tilde{s}_n, \tilde{S}_n, \tilde{w}_n \rangle)_{n \in \bar{\omega}} \in (\mathcal{L} \times \mathcal{L}[X] \times \mathbf{A}^{+})^{\bar{\omega}} \\
\ [\ S[s_2/X] \sim \tilde{s}_0 \parallel \tilde{S}_0[s_2/X] \wedge \\
\quad \forall n \geq 1[\ (\tilde{w}_n \setminus \tau) = (c_{n-1}) \wedge \tilde{s}_{n-1} \parallel \tilde{S}_{n-1}[s_2/X] \xrightarrow{\tilde{w}_n}_{*b} \tilde{s}_n \parallel \tilde{S}_n[s_2/X]\ ]]].
\end{array}
\tag{31}
$$

Let $(c_n)_{n \in \omega} \in \mathbf{C}^{\omega}$, $w \in \mathbf{A}^{\omega}$ such that $(w \setminus \tau) = (c_n)_{n \in \omega}$ and $S[s_1/X] \xrightarrow{w}_{*}$.

First, let $\tilde{s}_0 \equiv D$ and $\tilde{S}_0 \equiv X$.

Next, let us define $(\langle \tilde{s}_n, \tilde{S}_n, \tilde{w}_n \rangle)_{n \in \bar{\omega}}$ inductively as follows: For $n \in \omega$, suppose $\tilde{s}_n$ and $\tilde{S}_n$ have been defined.

If $\tilde{s}_n = \perp$ or $\tilde{S}_n = \perp$, let $\langle \tilde{s}_{n+1}, \tilde{S}_{n+1}, \tilde{w}_{n+1} \rangle = \langle \perp, \perp, \perp \rangle$.

Otherwise, putting

$$
\begin{array}{rl}
N = & \{ \langle \tilde{s}, \tilde{S}, \tilde{w} \rangle \in \mathcal{L} \times \mathcal{L}[X] \times \mathbf{A}^{+} : (\tilde{w} \setminus \tau) = (c_{n+1}) \wedge \\
& \quad \exists w''[\ \theta_{\perp}((c_{n+1+i})_{i \in \omega}) = \theta_{\perp}(w'') \wedge \tilde{s}_n \parallel \tilde{S}_n[s_2/X] \xrightarrow{\tilde{w}}_{*b} \tilde{s} \parallel \tilde{S}[s_2/X] \\
& \qquad \wedge\ \tilde{s} \parallel \tilde{S}[s_1/X] \xrightarrow{w''}_{*}\ ]\},
\end{array}
$$

let

$$
\langle \tilde{s}_{n+1}, \tilde{S}_{n+1}, \tilde{w}_{n+1} \rangle = \begin{cases} \langle \perp, \perp, \perp \rangle & \text{if } N = \emptyset, \\ \text{any element of } N & \text{otherwise.} \end{cases}
$$

It can be shown by induction on n using Lemma 6, that

$$
\begin{array}{l}
\forall n \geq 1[\ \langle \tilde{s}_{n-1}, \tilde{S}_{n-1} \rangle \in \mathcal{L} \times \mathcal{L}[X] \wedge \tilde{w}_n \in \mathbf{A}^{+} \wedge (\tilde{w}_n \setminus \tau) = (c_{n-1}) \\
\qquad \wedge\ \tilde{s}_{n-1} \parallel \tilde{S}_{n-1}[s_2/X] \xrightarrow{\tilde{w}_n}_{*b} \tilde{s}_n \parallel \tilde{S}_n\ ].
\end{array}
$$

Thus one has (31). $\blacksquare$

Finally, let us prove (26) (iv). In order to prove it, the following lemma is employed.

Lemma 8 *Let $s_1, s_2 \in \mathcal{L}$ such that $\mathcal{C}_{\mathrm{RF}}(s_1) = \mathcal{C}_{\mathrm{RF}}(s_2)$. Then for $S \in \mathcal{L}[X]$, the following holds:*

$$
\begin{array}{l}
S[s_1/X] \xrightarrow{\tau^{\omega}}_{*} \\
\quad \Rightarrow \exists \tilde{w} \in \{\tau\}^{+}, \exists \tilde{s} \in \mathcal{L}, \exists \tilde{S} \in \mathcal{L}[X][\ \tilde{s} \parallel \tilde{S}[s_1/X] \xrightarrow{\tau^{\omega}}_{*} \wedge S[s_2/X] \xrightarrow{\tilde{w}}_{*b} \tilde{s} \parallel \tilde{S}[s_2/X]\ ]. \quad \blacksquare
\end{array}
$$

Proof. This lemma is proved by induction on the number k such that $\exists s_1'[\ \vdash_{(k)} \tilde{s} \parallel \tilde{S}[s_1/X] \xrightarrow{\tau} s_1' \wedge s_1' \xrightarrow{\tau^{\omega}}\ ]$, using Lemma 6. $\blacksquare$

By means of this lemma, (26) (iv) is proved in a similar fashion to the way Lemma 7 is proved by means of Lemma 6. Thus, one has:

Lemma 9 *Let $s_1, s_2 \in \mathcal{L}$ such that $\mathcal{C}_{\mathrm{RF}}(s_1) = \mathcal{C}_{\mathrm{RF}}(s_2)$, and $S \in \mathcal{L}[X]$. Then, (26) (iv) holds, i.e.,* $\mathrm{Div}(S[s_1/X]) = \mathrm{Div}(S[s_2/X])$. $\blacksquare$