Safe Composition of Systems of Communicating Finite State Machines (*Full Version*)

Franco Barbanera
Dipartimento di Matematica e Informatica
University of Catania
franco.barbanera@unict.it

Rolf Hennicker
Institute for Informatics
LMU Munich
hennicke@pst.ifi.lmu.de

The *Participants-as-Interfaces* (PaI) approach to system composition suggests that participants of a system may be viewed as interfaces. Given a set of systems, one participant per system is chosen to play the role of an interface. When systems are composed, the interface participants are replaced by *gateways* which communicate to each other by forwarding messages. The PaI-approach for systems of asynchronous communicating finite state machines (CFSMs) has been exploited in the literature for binary composition only, with a (necessarily) unique forwarding policy. In this paper we consider the case of multiple system composition when forwarding gateways are not uniquely determined and their interactions depend on specific *connection policies* complying with a *connection model*. We represent connection policies as CFSM systems and prove that a bunch of relevant communication properties (deadlock-freeness, reception-error-freeness, etc.) are preserved by *PaI multicomposition*, with the proviso that also the used connection policy does enjoy the communication property taken into account.

1 Introduction

Concurrent/Distributed systems are hardly – especially nowadays – stand-alone entities. They are part of “jigsaws” never completely finished. Either in their design phase or after their deployment, they should be considered as *open* and ready for interaction with their environment, and hence with other systems. The possibility of extending and improving their functional and communication capabilities by composing them with other systems is also a crucial means against their obsolescence. Compositional mechanisms and techniques are consequently an important subject for investigation. As mentioned in [3], system composition investigations should focus on three relevant features of these mechanisms/techniques:

- *Conservativity:* They should alter as little as possible the single systems we compose.
- *Flexibility:* They should not be embedded into the systems we compose, i.e. they should be “system independent”. In particular, they should allow to consider **any** system as potentially **open**.
- *Safety:* Relevant properties of the single systems should not be “broken” by composition.

A fairly general and abstract approach to binary composition of systems was proposed in [1] and dubbed afterwards *Participants-as-Interfaces* (PaI). Roughly, the composition is achieved by transforming two selected participants – one per system, say h and k, – into coupled forwarders (gateways), provided the participants exhibit “compatible” behaviours. The graphics in Fig. 1 illustrates the PaI idea for the binary case. If interface participant h of the first system S_1 can receive a message a from some participant of S_1 and interface participant k of the second system S_2 can send a to some participant of S_2, then the gateway replacing the first interface (also called h) will forward the received message to the gateway for k. How PaI works for multicomposition of systems will be illustrated in Section 2. It is

To appear in EPTCS.

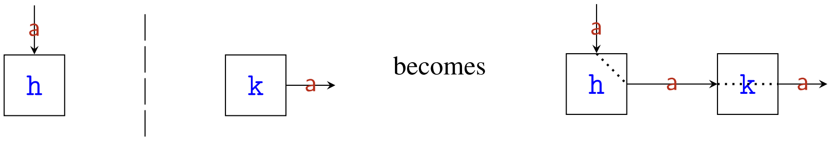


Figure 1: The PaI idea for binary composition

worth remarking that the PaI approach to system composition does not expect any particular condition to be satisfied by a single participant in order to be used as an interface.

Conservativity as well as *flexibility* are definitely features of the PaI composition idea. Conservativity holds since all participants not acting as interfaces remain untouched and flexibility holds since, in principle, any participant can play the role of an interface. This fact is independent of the concrete formalism used for protocol descriptions and system designs/implementations. *Safety*, instead, can be checked only once we take into account a specific formalism. Such checks were carried out in a number of papers where two relevant formalisms for the description and verification of concurrent communicating systems were considered: MultiParty Session Types (MPST) [20, 21] and Communicating Finite State Machines (CFSM) [11]. Safety of the binary PaI approach was investigated for MPST in [4], where a synchronous communication model was considered. The PaI approach to multicomposition for MPST has been exploited in [3, 2], again for synchronous communications. In particular, in [2], a restricted notion of multiple connections in a client-server setting has been considered. For the synchronous MPST formalism used in those papers, PaI proved to be safe. The binary PaI approach for safety in systems of (standard) asynchronous CFSMs was taken into account in [1], whereas safety of PaI for a synchronous version of the CFSM formalism was investigated in [5, 6, 7], again for binary composition.

Contributions. In the present paper we investigate safety of PaI multicomposition for the asynchronous formalism of CFSMs. For this purpose we reuse the PaI multicomposition idea of [3] but realise it – instead of the synchronous MPST framework – in the asynchronous CFSM setting which needs completely different design and proof techniques. At the same time we go beyond the binary composition of asynchronous CFSMs of [1] and study multicomposition of CFSM systems. Clearly this goes also beyond the aforementioned papers [5, 6, 7] dealing with binary composition of synchronous CFSMs. In particular, in the asynchronous case different communication properties, like freeness of unspecified receptions, are relevant.

A crucial role in our approach to multicomposition is played by *connection policies* which can be individually chosen by the system designer on the basis of a given concrete *connection model*. A connection model describes architectural aspects of compositions. It specifies which forwarding links between interface roles of different systems are meaningful from a static perspective. The concrete behavioural instantiation of such links, in terms of which message of an interface role, say h, is forwarded in which state of h to which interface role of another system, is determined by a connection policy which therefore also determines the construction of gateway CFSMs. The *multicomposition* of n systems of CFSMs is then simply defined by taking all CFSMs of the single systems but replacing each CFSM of an interface participant by its gateway CFSM. The use of connection models is methodologically important since it is more likely that a connection policy complying with a connection model will satisfy desired communication properties. Otherwise connection models are not relevant to our proofs.

In other words, we assume the existence of some connection model with which the connection policy used for the multicomposition is compliant. However, the specifics of the connection model are irrelevant for the safety results.

We show that a number of relevant communication properties (deadlock-freeness, orphan-message freeness, unspecified-reception freeness, and progress) are preserved by PaI multicomposition of CFSM systems whenever the particular property is satisfied also by the connection policy used, which is formalised as a CFSM system itself. Apart from orphan-message-freeness preservation we need, however, an additional assumption which requires that interface participants do not have a state with at least one outgoing output action and one outgoing input action, a condition referred to in the literature as *no-mixed-state* [14]. We shall provide counterexamples illustrating the role played by the no-mixed-state condition in guaranteeing safety of composition. In contrast with deadlock-freeness, the stronger property of lock-freeness will be shown (by means of a counterexample) not to be preserved in general, even in absence of of mixed-states.

Outline. The main ideas underlying PaI multicomposition are intuitively described in Section 2. In Section 3 we recall the definitions of communicating finite state machine, communicating system and their related notions. There we also provide the definitions of a number of relevant communication properties. In Section 4, PaI multicomposition is formally defined on the basis of the definitions of connection policy and gateway. Our main results are presented in Section 5 including counterexamples spotting the role of the no-mixed-state condition and a counterexample for lock-freeness preservation. Section 6 concludes with a brief summary, by pointing out a few more approaches to system composition, and with hints for future work.

2 The PaI Approach to Multicomposition

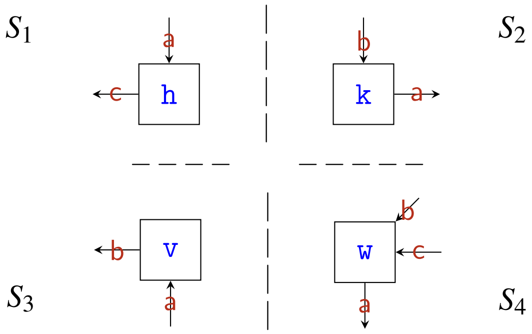


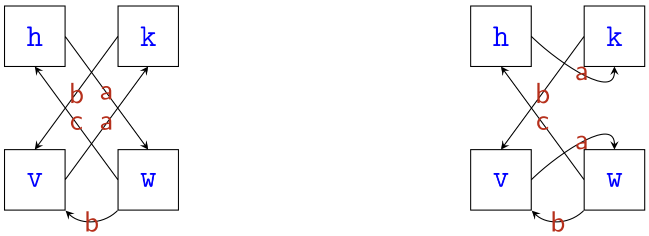
Figure 2: Four interface participants

In order to illustrate the idea underlying *PaI multicomposition*[1], we consider an example of [3] with four systems S_1, S_2, S_3 and S_4. As shown in Fig. 2, we have selected for each system one participant as an interface, named h, k, v and w. As in Fig. 1, we consider here only static aspects abstracting from dynamic issues, like the logical order of the exchanged messages, whose representation depends on the chosen formalism.

Following the PaI approach, the composition of the four systems above consists in replacing the participants h, k, v and w, chosen as interfaces, by gateways. Note that a message, like a in S_1 sent to h, could be forwarded (unlike the binary case) to different other gateways. This means that a *connection policy* has to be set up in order to appropriately define the gateways. Such a policy primarily depends on which partner is chosen for the current message to be exchanged.

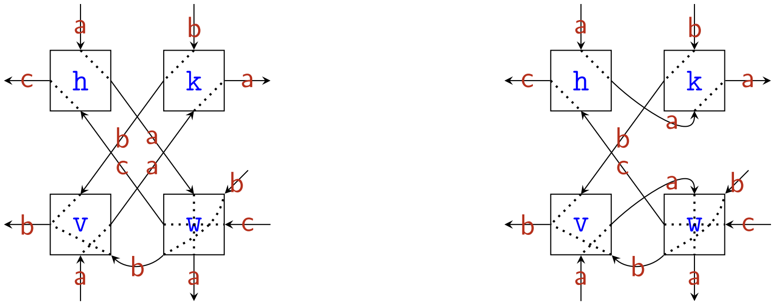
For what concerns the present example, one could decide that message a received by h has to be forwarded to w; the a received by v to k; the b received by k and w to v; the c received by w to h. Another possible choice could be similar to the previous one but for the forwarding of the messages a: the one received by h could be forwarded now to k whereas the one received by v could be forwarded to w. Such different "choices of partners", that we formalise by introducing the notion of *connection model*, can be graphically represented, respectively, by Choice A and Choice B in Fig. 3.

[1] It is of course possible to compose, two by two, several systems using binary composition, but in that way – by looking at systems as vertices and gateway connections as undirected edges – we can get only tree-like structures of systems.



(Choice A) (Choice B)

Figure 3: Two possible choices of partners.



(Using Choice A) (Using Choice B)

Figure 4: Two possible PaI multicompositions via gateways

The architecture of the resulting composed systems, according to the particular choices of partners (i.e. connection models), are represented by the diagrams in Fig. 4.

In both drawings of Fig. 4, the names h, k, etc. do now represent gateways. It is important to see that even if the original CFSMs for the participants in the single systems, like the CFSM for v in S_3, are given, the connection models and the drawings in Fig. 4 do not always provide what a gateway CFSM, modelling the dynamic forwarding strategy, should look like. This can be illustrated by looking at message b and participant v. No matter whether we consider Choice A or Choice B it is not determined when the gateway for v will accept b from k and when from w. For instance, a message b from w could be accepted by v only after two b's are received from k. Therefore, a given choice of partners needs, in general, to be "refined" – according to the formalism taken into account – into a specific *connection policy* taking care of the dynamic choice of partners.

This PaI approach to multicomposition has been exploited in [3] for a MPST formalism with synchronous communications. We are now going to realise PaI multicomposition in the context of CFSM systems with asynchronous communications.

3 Systems of Communicating Finite State Machines

Communicating Finite State Machines (CFSMs) is a widely investigated formalism for the description and analysis of distributed systems, originally proposed in [11]. CFSMs are a variant of finite state I/O-automata that represent processes which communicate by asynchronous exchanges of messages via FIFO channels. We now recall (partly following [14, 16, 23, 1]) the definitions of CFSM and system of

CFSMs.

We assume given a countably infinite set $\mathbf{P}_{\mathfrak{U}}$ of participant names (ranged over by $\mathtt{p}, \mathtt{q}, \mathtt{r}, \mathtt{h}, \mathtt{k}, \ldots$) and a countably infinite alphabet $\mathbb{A}_{\mathfrak{U}}$ of messages (ranged over by a, b, c, l,$\mathsf{m}, \ldots$).

Definition 3.1 (CFSM)**.** *Let* $\mathbf{P}$ *and* $\mathbb{A}$ *be finite subsets of* $\mathbf{P}_{\mathfrak{U}}$ *and* $\mathbb{A}_{\mathfrak{U}}$ *respectively.*

i) The set $C_{\mathbf{P}}$ *of* channels *over* $\mathbf{P}$ *is defined by* $C_{\mathbf{P}} = \{\mathtt{pq} \mid \mathtt{p}, \mathtt{q} \in \mathbf{P}, \mathtt{p} \neq \mathtt{q}\}$

ii) The set $Act_{\mathbf{P},\mathbb{A}}$ *of* actions *over* $\mathbf{P}$ *and* $\mathbb{A}$ *is defined by* $Act_{\mathbf{P},\mathbb{A}} = C_{\mathbf{P}} \times \{!, ?\} \times \mathbb{A}$
The subject *of an output action* $\mathtt{pq}!\mathsf{m}$ *and of an input action* $\mathtt{qp}?\mathsf{m}$ *is* $\mathtt{p}$.

iii) A communicating finite-state machine over $\mathbf{P}$ and $\mathbb{A}$ *is a finite transition system given by a tuple*

$$M = (Q, q_0, \mathbb{A}, \delta)$$

where Q *is a finite set of states,* $q_0 \in Q$ *is the initial state, and* $\delta \subseteq Q \times Act_{\mathbf{P},\mathbb{A}} \times Q$ *is a set of transitions such that all the actions have the same subject, to which we refer as the* name *of* M.

We shall write $M_{\mathtt{p}}$ to denote a CFSM with name $\mathtt{p}$. Where no ambiguity arises we shall refer to a CFSM by its name.

Notice that the above definition of CFSM is generic with respect to the underlying sets $\mathbf{P}$ and $\mathbb{A}$. This is necessary, since we shall not deal with a single system of CFSMs but with an arbitrary number of systems of CFSMs that can be *composed*. We shall write C and Act instead of $C_{\mathbf{P}}$ and $Act_{\mathbf{P},\mathbb{A}}$ when no ambiguity can arise. We assume $l, l', \ldots$ to range over Act; $\varphi, \varphi', \ldots$ to range over Act^* (the set of finite words over Act), and $w, w', \ldots$ to range over $\mathbb{A}^*$ (the set of finite words over $\mathbb{A}$). The symbol ε ($\notin \mathbb{A} \cup Act$) denotes the empty word and $\mid v \mid$ the lenght of a word $v \in Act^* \cup \mathbb{A}^*$.

The transitions of a CFSM are labelled by actions; a label $\mathtt{sr}!\mathsf{a}$ represents the asynchronous sending of message a from machine $\mathtt{s}$ to $\mathtt{r}$ through channel $\mathtt{sr}$ and, dually, $\mathtt{sr}?\mathsf{a}$ represents the reception (consumption) of a by $\mathtt{r}$ from channel $\mathtt{sr}$.

Given a CFSM $M = (Q, q_0, \mathbb{A}, \delta)$, we also define

$$\mathsf{in}(M) = \{\mathsf{a} \mid (\_, \_\_?\mathsf{a}, \_) \in \delta\} \qquad \text{and} \qquad \mathsf{out}(M) = \{\mathsf{a} \mid (\_, \_\_!\mathsf{a}, \_) \in \delta\}.$$

If M is a CFSM with name $\mathtt{p}$, we also write $\mathsf{in}(\mathtt{p})$ for $\mathsf{in}(M)$ and $\mathsf{out}(\mathtt{p})$ for $\mathsf{out}(M)$. Note that, in concrete examples, the name of a CFSM together with its input and output messages can be graphically depicted as in Fig. 2.

A state $q \in Q$ with no outgoing transition is *final*; q is a *sending* (resp. *receiving*) state if it is not final and all outgoing transitions are labelled with sending (resp. receiving) actions; q is a *mixed* state if there are at least two outgoing transitions such that one is labelled with a sending action and the other one is labelled with a receiving action.

A *communicating system*, called "protocol" in [11], is a finite set of CFSMs. In [14, 16, 23] the names of the CFSMs in a system are called *roles*. In the present paper we call them *participants*.

The dynamics of a system is formalised as a transition relation on configurations, where a configuration is a pair of tuples: a tuple of states of the machines in the system and a tuple of buffers representing the content of the channels.

Definition 3.2 (Communicating system and configuration)**.** *Let* $\mathbf{P}$ *and* $\mathbb{A}$ *be as in Def. 3.1.*

i) A communicating system (CS) over $\mathbf{P}$ and $\mathbb{A}$ *is a set* $S = (M_{\mathtt{p}})_{\mathtt{p} \in \mathbf{P}}$ *where for each* $\mathtt{p} \in \mathbf{P}$, $M_{\mathtt{p}} = (Q_{\mathtt{p}}, q_{0\mathtt{p}}, \mathbb{A}, \delta_{\mathtt{p}})$ *is a CFSM over* $\mathbf{P}$ *and* $\mathbb{A}$.

ii) A configuration *of a system* S *is a pair* $s = (\vec{q}, \vec{w})$ *where*

$$\vec{q} = (q_{\mathtt{p}})_{\mathtt{p} \in \mathbf{P}} \text{ with } q_{\mathtt{p}} \in Q_{\mathtt{p}}, \qquad \text{and} \qquad \vec{w} = (w_{\mathtt{pq}})_{\mathtt{pq} \in C} \text{ with } w_{\mathtt{pq}} \in \mathbb{A}^*.$$

The component $\vec{q}$ *is the* control state *of the system and* $q_{\mathtt{p}} \in Q_{\mathtt{p}}$ *is the* local state *of machine* $M_{\mathtt{p}}$. *The component* $\vec{w}$ *represents the state of the channels of the system and* $w_{\mathtt{pq}} \in \mathbb{A}^*$ *is the state of the channel* $\mathtt{pq}$, *i.e. the messages sent from* $\mathtt{p}$ *to* $\mathtt{q}$. *The initial configuration of* S *is* $s_0 = (\vec{q}_0, \vec{\varepsilon})$ *with* $\vec{q}_0 = (q_{0_{\mathtt{p}}})_{\mathtt{p} \in \mathbf{P}}$.

In the following we shall often denote a communicating system $(M_{\mathtt{p}})_{\mathtt{p} \in \{\mathtt{r}_i\}_{i \in I}}$ by $(M_{\mathtt{r}_i})_{i \in I}$.

Definition 3.3 (Reachable configuration)**.** *Let* S *be a communicating system over* $\mathbf{P}$ *and* $\mathbb{A}$, *and let* $s = (\vec{q}, \vec{w})$ *and* $s' = (\vec{q'}, \vec{w'})$ *be two configurations of* S. *Configuration* s' is reachable from s by firing a transition *with action* l, *written* $s \xrightarrow{l} s'$, *if there is* $\mathsf{a} \in \mathbb{A}$ *such that one of the following conditions holds:*

1. $l = \mathtt{sr}!\mathsf{a}$ *and* $(q_{\mathtt{s}}, l, q'_{\mathtt{s}}) \in \delta_{\mathtt{s}}$ *and*
   a) *for all* $\mathtt{p} \neq \mathtt{s}: \ q'_{\mathtt{p}} = q_{\mathtt{p}}$ *and*
   b) $w'_{\mathtt{sr}} = w_{\mathtt{sr}} \cdot \mathsf{a}$ *and for all* $\mathtt{pq} \neq \mathtt{sr}: \ w'_{\mathtt{pq}} = w_{\mathtt{pq}}$;
2. $l = \mathtt{sr}?\mathsf{a}$ *and* $(q_{\mathtt{r}}, l, q'_{\mathtt{r}}) \in \delta_{\mathtt{r}}$ *and*
   a) *for all* $\mathtt{p} \neq \mathtt{r}: \ q'_{\mathtt{p}} = q_{\mathtt{p}}$ *and*
   b) $w_{\mathtt{sr}} = \mathsf{a} \cdot w'_{\mathtt{sr}}$ *and for all* $\mathtt{pq} \neq \mathtt{sr}: \ w'_{\mathtt{pq}} = w_{\mathtt{pq}}$.

We write $s \rightarrow s'$ *if there exists* l *such that* $s \xrightarrow{l} s'$ *and we write* $s \not\rightarrow$ *if no* s' *and no* l *exist with* $s \xrightarrow{l} s'$. *As usual, we denote the reflexive and transitive closure of* $\rightarrow$ *by* $\rightarrow^*$. *The set of* reachable configurations *of* S *is* $\mathsf{RC}(S) = \{s \mid s_0 \rightarrow^* s\}$.

According to the above definition, communication happens via buffered channels following the FIFO principle.

The overall behaviour of a system can be described (at least) by the traces of configurations that are reachable from a distinguished initial one. Configurations may exhibit some pathological properties, like various forms of *deadlock* or *progress violation*, channels containing messages that will never be consumed (*orphan messages*) or just sent to a participant who is expecting another message to come (*unspecified receptions*). The goal of the analysis of communicating systems is to check whether such kinds of configurations are reachable or not. Although the desirable system properties are undecidable in general [11], sufficient conditions are known that are effectively checkable relying, for instance, on half-duplex communication [14], on the form of network topologies [15], or on synchronous compatibility checking [18].

We formalise now a number of relevant communication properties for systems of CFSMs that we shall deal with in the present paper.

Definition 3.4 (Communication properties)**.** *Let* S *be a communicating system, and let* $s = (\vec{q}, \vec{w})$ *be a configuration of* S.

i) *s is a* deadlock configuration *of* S *if* $\vec{w} = \vec{\varepsilon}$ *and* $\forall \mathtt{p} \in \mathbf{P}.\ q_{\mathtt{p}}$ *is a receiving state.*
   I.e. all buffers are empty, but all machines are waiting for a message.
   We say that S *is* deadlock-free *whenever, for any* $s \in \mathsf{RC}(S)$, s *is not a deadlock configuration.*

ii) *s is an* orphan-message configuration *of* S *if* $\forall \mathtt{p} \in \mathbf{P}.\ q_{\mathtt{p}}$ *is final and* $\vec{w} \neq \vec{\varepsilon}$.
   I.e. each machine is in a final state, but there is still at least one non-empty buffer. We say that S *is* orphan-message free *whenever, for any* $s \in \mathsf{RC}(S)$, s *is not an orphan-message configuration.*

iii) *s is an* unspecified reception configuration *of* S *if* $\exists \mathtt{r} \in \mathbf{P}$ *such that*
   a) $q_{\mathtt{r}}$ *is a receiving state; and*

b) $\forall \mathtt{s} \in \mathbf{P}.[\,(q_{\mathtt{r}}, \mathtt{sr}?\mathsf{a}, q'_{\mathtt{r}}) \in \delta_{\mathtt{r}} \implies (|w_{\mathtt{sr}}| > 0 \;\wedge\; w_{\mathtt{sr}} \notin \mathsf{a} \cdot \mathbb{A}^*)\,]$.

I.e. there is a receiving state $q_{\mathtt{r}}$ which is prevented from receiving any message from any of its buffers. (In other words, in each channel $\mathtt{sr}$ *from which role* $\mathtt{r}$ *could consume there is a message which cannot be received by* $\mathtt{r}$ *in state $q_{\mathtt{r}}$.) We say that S is* reception-error free *whenever, for any $s \in \mathsf{RC}(S)$, s is not an unspecified reception configuration.*

iv) *S satisfies the* progress property *if for all $s = (\vec{q}, \vec{w}) \in \mathsf{RC}(S)$, either there exists s' such that $s \rightarrow s'$ or* $(\forall \mathtt{p} \in \mathbf{P}.\ q_{\mathtt{p}}$ *is final*$)$.

v) *s is a* $\mathtt{p}$-lock configuration *of S if $\mathtt{p} \in \mathbf{P}$, $q_{\mathtt{p}}$ is a receiving state and*

$\mathtt{p}$ does not appear as subject in any label of any transition sequence from s

i.e. $\mathtt{p}$ remains stuck in all possible transition sequences from s. We say that S is lock-free *whenever, for each $\mathtt{p} \in \mathbf{P}$ and each $s \in \mathsf{RC}(S)$, s is not a $\mathtt{p}$-lock configuration.*

Note that progress property (iv) implies deadlock-freeness. Moreover, an unspecified reception configuration is trivially a $\mathtt{p}$-lock for some $\mathtt{p}$. This immediately implies that lock-freeness implies reception-error-freeness. It is also straightforward to check that lock-freeness does imply both deadlock-freeness and progress. The other properties are mutually independent.

The above definitions of communication properties (i)–(iv) are the same as the properties considered in [16], though the above formulation of progress is slightly simpler but equivalent to the one in [16]. The notions of orphan message and unspecified reception are also the same as in [23]. The same notions of deadlock and unspecified reception are given in [14] and inspired by [11]. The deadlock notions in [11] and [23] coincide with [14] and [16] if the local CFSMs have no final states. Otherwise deadlock in [23] is weaker than deadlock above. A still weaker notion of deadlock configuration, and hence a stronger notion of deadlock-freeness, has been suggested in [27]. This deadlock notion has been formally related to the above communication properties in [1].

4 PaI Multicomposition of Communicating Systems

As described in Section 2, the PaI approach to multicomposition of systems consists in replacing, in each to-be-composed system, one participant identified as an interface by a forwarder (that we dub "gateway"). Any participant in a system, say $\mathtt{h}$, can be considered as an interface. This means that we can look at the CFSM $\mathtt{h}$ as an abstract description of what the system expects from a number of "outer" systems (the environment) through their respective interfaces. Hence, any message received by $\mathtt{h}$ from another participant $\mathtt{p}$ of the system (to which $\mathtt{h}$ belongs) is interpreted as a message to be forwarded to some other interface $\mathtt{h}'$ among the available ones. Conversely, any message sent from $\mathtt{h}$ to another participant $\mathtt{p}$ of the system (to which $\mathtt{h}$ belongs) is interpreted as a message to be received from some other interface $\mathtt{h}'$ and to be forwarded to $\mathtt{p}$.

In order to clarify the notions introduced in this section, we present below an example from [3], "implemented" here in the CFSM formalism.

Example 4.1 (Working example)**.** Let us consider the following four systems[2]:

System-1 with participants $\mathtt{h}_1$ and $\mathtt{p}$.

Participant $\mathtt{h}_1$ controls the entrance of customers in a mall (via some sensor). As soon as a customer enters, $\mathtt{h}_1$ sends a message start to the participant $\mathtt{p}$ which controls a display for advertisements.

[2] For the sake of simplicity, the example considers only systems with two or three participants. Our definitions and results are of course independent of the number of participants in the single systems.

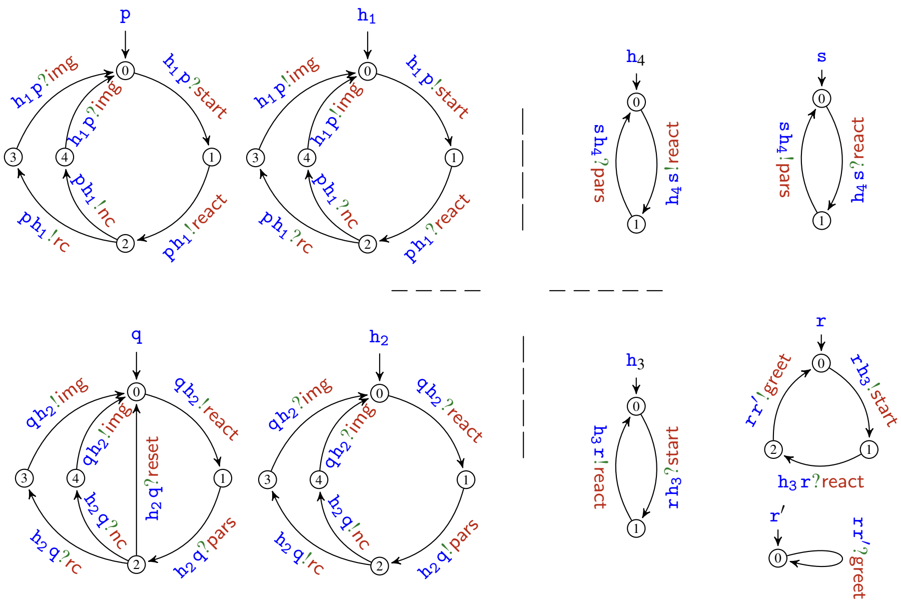


Figure 5: The four communicating systems formalising the systems of Example 4.1

On receiving the start message, $\mathtt{p}$ displays a general advertising image. Participant $\mathtt{p}$ does also control a sensor detecting emotional reactions as well as a card reader distinguishing regular from new customers. Such information, through the messages react, rc and nc is sent to $\mathtt{h}_1$. Using that information $\mathtt{h}_1$ sends to $\mathtt{p}$ a customised image, depending on the kind of the customer, through message img.

System-2 with participants $\mathtt{h}_2$ and $\mathtt{q}$.

Participant $\mathtt{h}_2$ controls an image display. Images are provided by participant $\mathtt{q}$ according to some parameters sent by $\mathtt{h}_2$ itself and depending on the reaction acquired by a sensor driven by $\mathtt{q}$. Images are chosen also in terms of the kind of customers, on the basis of their cards. Participant $\mathtt{q}$ is able to receive a reset message too, even if $\mathtt{h}_2$ cannot ever send it.

System-3 with participants $\mathtt{h}_3$, $\mathtt{r}$ and $\mathtt{r}'$.

Participant $\mathtt{r}$ controls a sensor detecting the entrance of people from a door. Once someone enters, a message start is sent by $\mathtt{r}$ to participant $\mathtt{h}_3$ which turns on a light. The reaction of who enters, detected by a sensor driven by $\mathtt{h}_3$, is sent back to $\mathtt{r}$ which, according to the reaction, communicates to $\mathtt{r}'$ the greeting to be broadcasted from the loudspeakers.

System-4 with participants $\mathtt{h}_4$ and $\mathtt{s}$.

Some sensors driven by Participant $\mathtt{h}_4$ acquire the first reactions of people getting into a hall adorned by several Christmas lights. Such reactions, sent to participant $\mathtt{s}$ through a message react, enable $\mathtt{s}$ to send to $\mathtt{h}_4$ a set of parameters (pars) allowing the latter to adjust the lights of the hall.

The behaviours of the participants of the above systems – assuming an asynchronous model of communication – can be formalised as CFSMs. So the systems above can be formalised as the following communicating systems

$$S_1 = (M_{\mathtt{x}})_{\mathtt{x}\in\{\mathtt{h}_1,\mathtt{p}\}} \quad S_2 = (M_{\mathtt{x}})_{\mathtt{x}\in\{\mathtt{h}_2,\mathtt{q}\}} \quad S_3 = (M_{\mathtt{x}})_{\mathtt{x}\in\{\mathtt{h}_3,\mathtt{r},\mathtt{r}'\}} \quad S_4 = (M_{\mathtt{x}})_{\mathtt{x}\in\{\mathtt{h}_4,\mathtt{s}\}}$$

as described, anticlockwise, in Figure 5. ◇

Notation: We use the following notation to denote the above set of communicating systems: $\{S_i\}_{i\in\{1,2,3,4\}}$ where $S_i = (M_{\mathtt{x}})_{\mathtt{x}\in\mathbf{P}_i}$ with $\mathbf{P}_1 = \{\mathtt{h}_1,\mathtt{p}\}$, $\mathbf{P}_2 = \{\mathtt{h}_2,\mathtt{q}\}$, $\mathbf{P}_3 = \{\mathtt{h}_3,\mathtt{r},\mathtt{r}'\}$ and $\mathbf{P}_4 = \{\mathtt{h}_4,\mathtt{s}\}$.

The composition of a set of systems relies on a selection of participants, one for each system, considered as interfaces.

Definition 4.2 (Interfaces). *Let $\{S_i\}_{i\in I}$ be a set of communicating systems such that, for each $i \in I$, $S_i = (M_{\mathtt{x}})_{\mathtt{x}\in\mathbf{P}_i}$, where the $\mathbf{P}_i$'s are pairwise disjoint. A set of participants $H = \{\mathtt{h}_i\}_{i\in I} \subseteq \bigcup_{i\in I}\mathbf{P}_i$ is a* set of interfaces *for $\{S_i\}_{i\in I}$ whenever, for each $i \in I$, $\mathtt{h}_i \in \mathbf{P}_i$. An interface $\mathtt{h}_i$ has* no mixed states *if the CFSM M_{h_i} in S_i has no mixed states.*

Example 4.3. We choose $\{\mathtt{h}_i\}_{i\in\{1,2,3,4\}}$ as set of interfaces for the communicating systems of Figure 5. ◇

We introduce now the notion of *connection model*[3], formalising what we have informally called "choice of partners" in Section 2. A connection model is intended to specify the structural (architectural) aspects of possible "reasonable" connections between interfaces of systems. Connection models should be provided before systems are composed since they help the system designer to avoid blatantly unreasonable compositions. Formally, a connection model is a set of *connections*, where a connection is a triple $(\mathtt{h},\mathtt{a},\mathtt{h}')$ in which $\mathtt{h}$ and $\mathtt{h}'$ are, respectively, interfaces of two systems, say S and S', and $\mathtt{a}$ is an input message for $\mathtt{h}$ and an output message for $\mathtt{h}'$. Being $\mathtt{a}$ an input for $\mathtt{h}$, this participant is supposed to receive $\mathtt{a}$ from the "inside" of S, i.e. from another participant of S. As previously mentioned, PaI multicomposition relies on the idea that $\mathtt{a}$ can be forwarded to the interface of some other system. The connection $(\mathtt{h},\mathtt{a},\mathtt{h}')$ hence specifies that $\mathtt{h}'$ is one of the possible interfaces $\mathtt{a}$ can be forwarded to. This is sound since $\mathtt{a}$ is an output of $\mathtt{h}'$, i.e. it is sent by $\mathtt{h}'$ to some participant of S'. The actual composition will then rely on gateways (forwarders) which comply with the connection model taken into account.

Definition 4.4 (Connection model). *Let $\{S_i\}_{i\in I}$ be a set of communicating systems and let H be a set of interfaces for it.*

i) *A* connection model for H *is a ternary relation* $\textsc{cm} \subseteq H \times \mathbb{A}_{\mathfrak{U}} \times H$ *such that, for each $\mathtt{h} \in H$ and $\mathtt{a} \in \mathbb{A}_{\mathfrak{U}}$,*

- $\mathtt{a} \in \mathsf{in}(\mathtt{h})$ *implies* $\exists\, \mathtt{h}' \in H$ *s.t.* $\mathtt{a} \in \mathsf{out}(\mathtt{h}')$ *and* $(\mathtt{h},\mathtt{a},\mathtt{h}') \in \textsc{cm}$
- $\mathtt{a} \in \mathsf{out}(\mathtt{h})$ *implies* $\exists\, \mathtt{h}' \in H$ *s.t.* $\mathtt{a} \in \mathsf{in}(\mathtt{h}')$ *and* $(\mathtt{h}',\mathtt{a},\mathtt{h}) \in \textsc{cm}$

where $\mathtt{h} \neq \mathtt{h}'$.
Elements of $\textsc{cm}$ *are called* connections. *In particular,* $(\mathtt{h},\mathtt{a},\mathtt{h}') \in \textsc{cm}$ *is called* connection for $\mathtt{a}$ (from $\mathtt{h}$ to $\mathtt{h}'$). *We also define* $\mathsf{Msg}(\textsc{cm}) = \{\mathtt{a} \mid (\_,\mathtt{a},\_) \in \textsc{cm}\}$ *and assume that any message* $\mathtt{a} \in \mathsf{Msg}(\textsc{cm})$ *occurs in one of the interfaces in H either as an input or as an output.*

ii) *A connection model* $\textsc{cm}$ *for H is* strong *if, for each $\mathtt{h} \in H$ and $\mathtt{a} \in \mathbb{A}_{\mathfrak{U}}$,*

- $\mathtt{a} \in \mathsf{in}(\mathtt{h})$ *implies* $\exists!\, \mathtt{h}' \in H$ *s.t.* $(\mathtt{h},\mathtt{a},\mathtt{h}') \in \textsc{cm}$
- $\mathtt{a} \in \mathsf{out}(\mathtt{h})$ *implies* $\exists!\, \mathtt{h}' \in H$ *s.t.* $(\mathtt{h}',\mathtt{a},\mathtt{h}) \in \textsc{cm}$.

where $\mathtt{h} \neq \mathtt{h}'$ *and the unique existential quantifier '$\exists!$' stands for "there exists exactly one".*

Connection models can be graphically represented by diagrams, like those used in Fig. 3.

[3] Such a notion was informally introduced in [3] in the setting of MultiParty Session Types.

Example 4.5 (Some connection models)**.** Let $H = \{\mathtt{h},\mathtt{k},\mathtt{v},\mathtt{w}\}$ be the set of interfaces for the systems $\{S_i\}_{i\in\{1,2,3,4\}}$ in Section 2. Fig. 3 represents the following connection models for H:

$$\begin{aligned} \textsc{cm}_{\mathrm{A}} &= \{(\mathtt{h},\mathsf{a},\mathtt{w}),(\mathtt{v},\mathsf{a},\mathtt{k}),(\mathtt{w},\mathsf{c},\mathtt{h}),(\mathtt{k},\mathsf{b},\mathtt{v}),(\mathtt{w},\mathsf{b},\mathtt{v})\} \\ \textsc{cm}_{\mathrm{B}} &= \{(\mathtt{h},\mathsf{a},\mathtt{k}),(\mathtt{v},\mathsf{a},\mathtt{w}),(\mathtt{w},\mathsf{c},\mathtt{h}),(\mathtt{k},\mathsf{b},\mathtt{v}),(\mathtt{w},\mathsf{b},\mathtt{v})\} \end{aligned}$$

Obviously, both connection models are not strong, because of the presence of the connections $(\mathtt{k},\mathsf{b},\mathtt{v})$ and $(\mathtt{w},\mathsf{b},\mathtt{v})$.

Let us now provide a connection model for the systems in Fig. 5 with set of interfaces $H = \{\mathtt{h}_i\}_{i\in\{1,2,3,4\}}$. First we determine $\mathsf{in}(\mathtt{h}_1) = \{\mathsf{react},\mathsf{nc},\mathsf{rc}\}$, $\mathsf{out}(\mathtt{h}_1) = \{\mathsf{img},\mathsf{start}\}$, $\mathsf{in}(\mathtt{h}_2) = \{\mathsf{react},\mathsf{img}\}$, $\mathsf{out}(\mathtt{h}_2) = \{\mathsf{nc},\mathsf{rc},\mathsf{pars}\}$, $\mathsf{in}(\mathtt{h}_3) = \{\mathsf{start}\}$, $\mathsf{out}(\mathtt{h}_3) = \{\mathsf{react}\}$, and $\mathsf{in}(\mathtt{h}_4) = \{\mathsf{pars}\}$, $\mathsf{out}(\mathtt{h}_4) = \{\mathsf{react}\}$. A connection model for H is

$$\begin{aligned} \textsc{cm} = \{&(\mathtt{h}_1,\mathsf{react},\mathtt{h}_4),(\mathtt{h}_3,\mathsf{start},\mathtt{h}_1),(\mathtt{h}_2,\mathsf{img},\mathtt{h}_1),(\mathtt{h}_1,\mathsf{nc},\mathtt{h}_2), \\ &(\mathtt{h}_1,\mathsf{rc},\mathtt{h}_2),(\mathtt{h}_4,\mathsf{pars},\mathtt{h}_2),(\mathtt{h}_2,\mathsf{react},\mathtt{h}_3)\} \end{aligned}$$

The representation of $\textsc{cm}$ is as in Fig. 6. Obviously, this connection model is strong. ◇

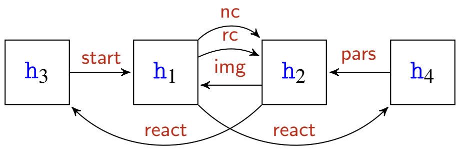


Figure 6: A connection model for the interfaces of Fig. 5.

When we have more than two systems to compose, the gateways are, in general, not uniquely determined. In order to produce gateways out of interfaces we need to decide which connection model we wish to take into account and how the interfaces do actually interact "complying" with the connection model. Once a connection model is selected, the forwarding strategy of the gateway is still not uniquely determined if the connection model is not strong. The reason is that in the case of at least two connectors with the same source or the same target, like $(\mathtt{k},\mathsf{b},\mathtt{v})$ and $(\mathtt{w},\mathsf{b},\mathtt{v})$ in Example 4.5, the gateway for $\mathtt{v}$ has a dynamic choice when to accept message b from $\mathtt{k}$ and when from $\mathtt{w}$. Therefore we need further (dynamic) information which will be provided by *connection policies*. A connection policy is itself a communicating system which describes the dynamic choice of partners among the possible gateways by respecting the constraints of (that is, complying with) the connection model. Technically, we first associate a set of CFSMs (the "local connection policy set") to each interface. Any element of this set specifies which communications to the "outside" are allowed in which state. Technically these communications are dual to the communications of its corresponding interface.

Definition 4.6 (Local Connection Policy Set)**.** *Let* $\textsc{cm}$ *be a connection model for a set of interfaces* H *and let* $\mathtt{h} \in H$ *with CFSM* $M_{\mathtt{h}} = (Q, q_0, \mathbb{A}, \delta)$.
The local connection policy set *of* $M_{\mathtt{h}}$ *w.r.t.* $\textsc{cm}$ *is the set of CFSMs* $\mathsf{LCPS}(M_{\mathtt{h}},\textsc{cm})$ *defined as follows:*

$$\mathsf{LCPS}(M_{\mathtt{h}},\textsc{cm}) = \{(\dot{Q},\dot{q}_0,\mathbb{A},\dot{\delta}) \mid \quad \dot{\delta} \textit{ is a minimal relation s.t. } (*) \textit{ and } (**)\}$$

where $\dot{Q} = \{\dot{q} \mid q \in Q\}$ *and*

$(*) = q \xrightarrow{\mathtt{rh}?\mathsf{a}} q' \in \delta$ *implies* $\exists \mathtt{p} \in H \setminus \{\mathtt{h}\}$ *s.t.* $\dot{q} \xrightarrow{\dot{\mathtt{h}}\dot{\mathtt{p}}!\mathsf{a}} \dot{q}' \in \dot{\delta}$ *and* $(\mathtt{h},\mathsf{a},\mathtt{p}) \in \textsc{cm}$,

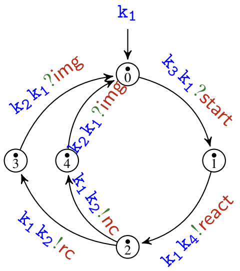
$(**) = q \xrightarrow{\mathtt{hr}!\mathsf{a}} q' \in \delta$ *implies* $\exists \mathtt{p} \in H \setminus \{\mathtt{h}\}$ *s.t.* $\dot{q} \xrightarrow{\dot{\mathtt{p}}\dot{\mathtt{h}}?\mathsf{a}} \dot{q}' \in \dot{\delta}$ *and* $(\mathtt{p},\mathsf{a},\mathtt{h}) \in \textsc{cm}$.

Notice that, in the above definition, each CFSM in $\mathsf{LCPS}(M_{\mathtt{h}}, \textsc{cm})$ has name $\dot{\mathtt{h}}$. Moreover, $\dot{q}$ (resp. $\dot{\mathtt{h}}$) is to be looked at as a "decoration" of the state q (resp. the name $\mathtt{h}$). This will enable us to immediately retrieve q (resp. $\mathtt{h}$) out of $\dot{q}$ (resp. $\dot{\mathtt{h}}$).

Notation: In the following, for the sake of readability, we shall write $\mathtt{k}$ (resp. $\mathtt{k}_i$) for $\dot{\mathtt{h}}$ (resp. $\dot{\mathtt{h}}_i$).

Local connection policy sets are finite, since they contain machines which only differ in the names of participants and these names belong to a finite set. Any element $(\dot{Q}, \dot{q}_0, \mathbb{A}, \dot{\delta})$ of $\mathsf{LCPS}(M_{\mathtt{h}}, \textsc{cm})$ does comply with the connection model $\textsc{cm}$, since it can only have transitions $\dot{q} \xrightarrow{\dot{\mathtt{h}}\mathtt{p}!\mathtt{a}} \dot{q}' \in \dot{\delta}$ with $(\mathtt{h}, \mathtt{a}, \mathtt{p}) \in \textsc{cm}$ and transitions $\dot{q} \xrightarrow{\mathtt{p}\dot{\mathtt{h}}?\mathtt{a}} \dot{q}' \in \dot{\delta}$ with $(\mathtt{p}, \mathtt{a}, \mathtt{h}) \in \textsc{cm}$, Moreover, $\mathsf{LCPS}(M_{\mathtt{h}}, \textsc{cm})$ is a singleton if the connection model $\textsc{cm}$ is strong.

Example 4.7 (An element of a local connection policy set)**.** Let $M_{\mathtt{h}_1}$ be the CFSM for the participant $\mathtt{h}_1$ of Example 4.1 and let $\textsc{cm}$ be the strong connection model for $H = \{\mathtt{h}_i\}_{i \in \{1,2,3,4\}}$ of Example 4.5. The CFSM on the right is the unique element of $\mathsf{LCPS}(M_{\mathtt{h}_1}, \textsc{cm})$. ◇

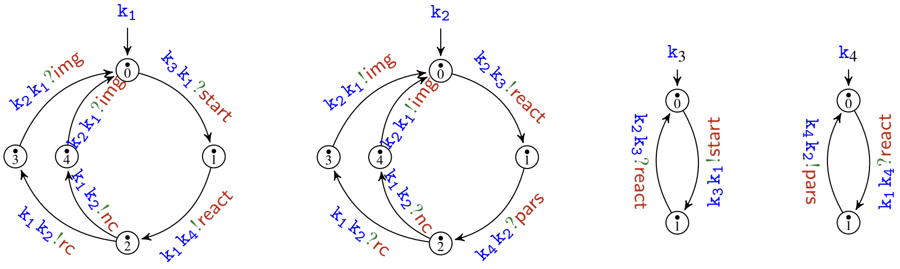


Given a connection model, a connection policy is obtained by choosing, for each interface, an element of its local connection policy set.

Definition 4.8 (Connection policy)**.** *Let $\{S_i\}_{i \in I}$ be a set of communicating systems such that, for each $i \in I$, $S_i = (M_{\mathtt{x}})_{\mathtt{x} \in \mathbf{P}_i}$, and let $\textsc{cm}$ be a connection model for a set of interfaces $H = \{\mathtt{h}_i\}_{i \in I}$. A* connection policy (for H) complying with $\textsc{cm}$ *is a communicating system $\mathbb{K} = (M_{\mathtt{k}_i})_{i \in I}$ such that, for each $i \in I$, $M_{\mathtt{k}_i} \in \mathsf{LCPS}(M_{\mathtt{h}_i}, \textsc{cm})$.*

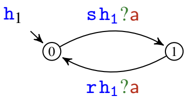
Connection policies are made of local connection policies which, due to the conditions $(*)$ and $(**)$ in Definition 4.6, are compliant with the given communication model $\textsc{cm}$. Consequently, in the above definition, the connection policy is said to be compliant with $\textsc{cm}$. If we dropped the two requirements $(*)$ and $(**)$ in Definition 4.6 we would get non-compliant connection policies.

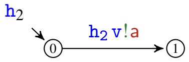
Example 4.9 (A connection policy)**.** The following four CFSMs constitute a connection policy for $H = \{\mathtt{h}_i\}_{i \in I}$ complying with $\textsc{cm}$, where the $M_{\mathtt{h}_i}$'s are as in Figure 5 and $\textsc{cm}$ is the connection model of Example 4.5.

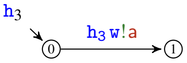


◇

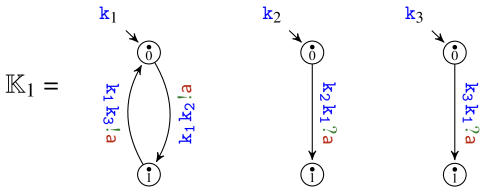
Remark 4.10. A connection model can be looked at as a static and abstract description of connection policies. In particular a connection model abstracts from the order of exchanged messages. As already pointed out above there may be several connection policies complying with a given connection model $\textsc{cm}$ if $\textsc{cm}$ is not strong. As an example assume given three systems with the following interfaces:

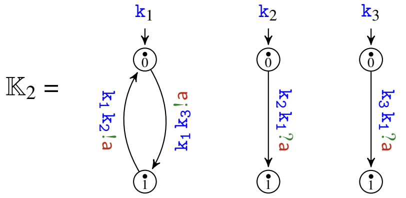






We can now consider the following (non-strong) connection model: $\textsc{cm} = \{(\mathtt{h}_1, \mathtt{a}, \mathtt{h}_2), (\mathtt{h}_1, \mathtt{a}, \mathtt{h}_3)\}$. It is easy to check that the connection policies $\mathbb{K}_1$ and $\mathbb{K}_2$ below do both comply with $\textsc{cm}$.





◇

By now we have almost all the necessary notions to formally define the PaI multicomposition of systems of communicating systems. The only missing piece is that of building the gateways using a connection policy.

We get a gateway essentially by transforming an interface $M_{\mathtt{h}}$ by inserting a fresh state in between any transition. Any input transition $q \xrightarrow{\mathtt{sh?a}} q'$ (resp. output transition $q \xrightarrow{\mathtt{hs!a}} q'$) of $M_{\mathtt{h}}$ is then transformed into two consecutive transitions

$$q \xrightarrow{\mathtt{sh?a}} \hat{q} \xrightarrow{\mathtt{hh'!a}} q' \qquad (\text{resp. } q \xrightarrow{\mathtt{h'h?a}} \hat{q} \xrightarrow{\mathtt{hs!a}} q')$$

where $\hat{q}$ is a fresh state and $\dot{q} \xrightarrow{\mathtt{kk'!a}} \dot{q}'$ (resp. $q \xrightarrow{\mathtt{k'k?a}} \dot{q}'$) belonging to the connection policy taken into account. In the formal definition below we distinguish the fresh states by superscripting them by the transition they are "inserted in between".

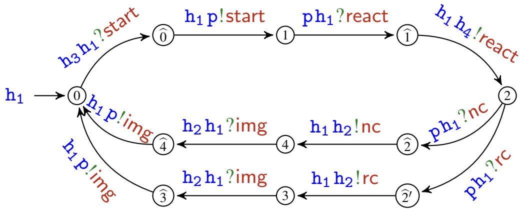
Definition 4.11 (Gateway)**.**
Assume given a connection model $\textsc{cm}$ *and two CFSMs* $M_{\mathtt{h}}$ *and* $M_{\mathtt{k}}$ *such that* $M_{\mathtt{h}} = (Q, q_0, \mathbb{A}, \delta)$ *and* $M_{\mathtt{k}} = (\dot{Q}, \dot{q}_0, \mathbb{A}, \dot{\delta}) \in \mathsf{LCPS}(M_{\mathtt{h}}, \textsc{cm})$. *The* gateway $M_{\mathtt{h}} \leftarrow\!\!\!\!\rho M_{\mathtt{k}}$ *obtained out of* $M_{\mathtt{h}}$ *and* $M_{\mathtt{k}}$ *is defined by*

$$M_{\mathtt{h}} \leftarrow\!\!\!\!\rho M_{\mathtt{k}} = (Q \cup \widehat{Q}, q_0, \mathbb{A}, \hat{\delta})$$

where
- $\widehat{Q} = \bigcup_{q \in Q} \{q^{(q,l,q')} \mid (q,l,q') \in \delta\}$,
- $\widehat{\delta} = \{(q, \mathtt{rh?a}, \widehat{q}), (\widehat{q}, \mathtt{hs!a}, q') \mid (q, \mathtt{hs!a}, q') \in \delta, (\dot{q}, \dot{\mathtt{r}}\mathtt{k?a}, \dot{q}') \in \dot{\delta},\ \widehat{q} = q^{(q,\mathtt{hs!a},q')}\}$
  $\cup \{(q, \mathtt{sh?a}, \widehat{q}), (\widehat{q}, \mathtt{hr!a}, q') \mid (q, \mathtt{sh?a}, q') \in \delta,\ (\dot{q}, \mathtt{k}\dot{\mathtt{r}}\mathtt{!a}, \dot{q}') \in \dot{\delta},\ \widehat{q} = q^{(q,\mathtt{sh?a},q')}\}$.

We refer to $\widehat{\delta}$ *as* $\widehat{\delta}_{\mathtt{h}}$ *whenever* $\mathtt{h}$ *is not clear from the context; similarly for* $\widehat{Q}$.

Example 4.12 (A gateway)**.** Let $M_{\mathtt{h}_1}$ be as in Example 4.1, and let $M_{\mathtt{k}_1}$ be as in the connection policy of Example 4.9. The gateway $M_{\mathtt{h}_1} \leftarrow\!\!\!\!\rho M_{\mathtt{k}_1}$ is as follows.



◇

Definition 4.13 (Composability)**.** *Let* $\{S_i\}_{i \in I}$ *be a set of communicating systems such that, for each* $i \in I$, $S_i = (M_{\mathtt{x}})_{\mathtt{x} \in \mathbf{P}_i}$. *Moreover, let* $H = \{\mathtt{h}_i\}_{i \in I}$ *be a set of interfaces for it. We say that* $\{S_i\}_{i \in I}$ *is* composable with respect to H *whenever the sets* $\mathbf{P}_i$*'s are pairwise disjoint.*

Let us now describe how systems are composed on the basis of a given connection policy.

Definition 4.14 (Multicomposition of communicating systems)**.** *Let $\{S_i\}_{i\in I}$ be a set of communicating systems composable with respect to $H = \{\mathtt{h}_i\}_{i\in I}$ and let $\mathbb{K} = (M_{\mathtt{k}_i})_{i\in I}$ be a connection policy complying with a connection model* CM *for H. The* multicomposition of $\{S_i\}_{i\in I}$ with respect to $\mathbb{K}$ *is the communicating system*

$$\mathscr{MC}(\{S_i\}_{i\in I},\mathbb{K}) = (M'_{\mathtt{p}})_{\mathtt{p}\in\bigcup_{i\in I}\mathbf{P}_i}$$

where

$$M'_{\mathtt{p}} = \begin{cases} M_{\mathtt{p}} & \textit{if } \mathtt{p} \notin \{\mathtt{h}_i\}_{i\in I} \\ M_{\mathtt{h}_i} \leftarrow\!\!\!\rho\, M_{\mathtt{k}_i} & \textit{if } \mathtt{p} = \mathtt{h}_i \textit{ with } i \in I \end{cases}$$

Note that the CFSMs of a composition are CFSMs over $\mathbf{P} = \bigcup_{i\in I}\mathbf{P}_i$ and $\mathbb{A} = \bigcup_{i\in I}\mathbb{A}_i$. Graphically, the architectural structure of a multicomposition via gateways can be shown as in Fig. 4.

5 On the Preservation of Communication Properties

The main result of the present paper is the safety of PaI multicomposition of CFSM systems for all communication properties of Definition 3.4 but lock-freeness. Apart from orphan-message-freeness we need the no-mixed-state assumption for interfaces to obtain the preservation results.

Theorem 5.1 (Safety of PaI multicomposition of CFSM systems)**.** *Let $\{S_i\}_{i\in I}$ be a set of communicating systems composable with respect to a set $H = \{\mathtt{h}_i\}_{i\in I}$ of interfaces with no mixed states (cf. Definition 4.2) and let $\mathbb{K}$ be a connection policy for H. Let $\mathscr{P}$ be either the property of* deadlock-freeness *or* reception-error-freeness *or* progress*. If $\mathscr{P}$ holds for each S_i with $i \in I$ and for $\mathbb{K}$, then $\mathscr{P}$ holds for $S = \mathscr{MC}(\{S_i\}_{i\in I},\mathbb{K})$. Moreover, the above holds also if the no-mixed-state condition is removed and $\mathscr{P}$ is* orphan-message-freeness.

Remark 5.2. The above result about safety of multicomposition is actually independent of a concrete connection model. Considering connection policies which comply with a connection model is, however, helpful at the design stage of the multicomposition and enhances the possibility of getting connection policies which satisfy communication properties and hence support the preservation of communication properties of the composed systems. ◇

Theorem 5.1 can be proved for each property $\mathscr{P}$ separately by contradiction. In particular by showing that if $\mathscr{P}$ does not hold for S then it does not hold either for one of the S_i's or for $\mathbb{K}$.

A key notion for the proofs is that of *projection* of a reachable configuration of the composed system to configurations of each of the single systems S_i and also of the connection policy $\mathbb{K}$. On this basis, the most important tool to get contradictions is the subsequent Proposition 5.4 which essentially shows that projections of reachable configurations involving no intermediate gateway states are reachable configurations again. The complete proofs of property preservations are provided in Appendix A. They are independent of the communication model $\mathbb{K}$ complies with.

Definition 5.3 (Configuration projections)**.** *Let $S = \mathscr{MC}(\{S_i\}_{i\in I},\mathbb{K})$ be as in Theorem 5.1 (but without no-mixed-state assumption). Let $s = (\vec{q},\vec{w}) \in \mathsf{RC}(S)$ where $\vec{q} = (q_{\mathtt{p}})_{\mathtt{p}\in\mathbf{P}}$ and $\vec{w} = (w_{\mathtt{pq}})_{\mathtt{pq}\in C_{\mathbf{P}}}$. For each $i \in I$, the projection $s_{|_\mathbf{i}}$ of s to S_i is defined by*

$$s_{|_\mathbf{i}} = (\vec{q}_{|_\mathbf{i}},\vec{w}_{|_\mathbf{i}})$$

where $\vec{q}_{|_\mathbf{i}} = (q_{\mathtt{p}})_{\mathtt{p}\in\mathbf{P}_i}$ and $\vec{w}_{|_\mathbf{i}} = (w_{\mathtt{pq}})_{\mathtt{pq}\in C_{\mathbf{P}_i}}$.
The projection $s_{|_\mathbb{K}}$ of $s = (\vec{q},\vec{w})$ to $\mathbb{K}$ is defined if $q_{\mathtt{h}_i} \notin \widehat{Q}_{\mathtt{h}_i}$ for each $i \in I$ and then

$$s_{|_\mathbb{K}} = (\vec{q}_{|_\mathbb{K}},\vec{w}_{|_\mathbb{K}})$$

where $\vec{q}_{|\mathbb{K}} = (p_{\mathtt{k}_i})_{i\in I}$ *is such that, for each* $i \in I$, $p_{\mathtt{k}_i} = \dot{q}_{\mathtt{h}_i}$ *(with* $\dot{q}_{\mathtt{h}_i}$ *being the "dotted decoration" of the local state* $q_{\mathtt{h}_i}$*) and where* $\vec{w}_{|\mathbb{K}} = (w'_{\mathtt{pq}})_{\mathtt{p},\mathtt{q}\in\{\mathtt{k}_i\}_{i\in I},\mathtt{p}\neq\mathtt{q}}$ *is such that, for each pair* $i, j \in I$ *with* $i \neq j$, $w'_{\mathtt{k}_i\mathtt{k}_j} = w_{\mathtt{h}_i\mathtt{h}_j}$.

Proposition 5.4 (On reachability of projections)**.** *Let* $s = (\vec{q},\vec{w}) \in \mathsf{RC}(S)$.

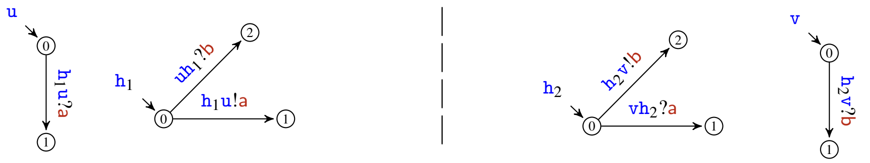
i) For each $i \in I$, $(q_{\mathtt{h}_i} \notin \widehat{Q_{\mathtt{h}_i}} \implies s_{|_\mathbf{i}} \in \mathsf{RC}(S_i))$;

ii) $(q_{\mathtt{h}_i} \notin \widehat{Q_{\mathtt{h}_i}}$ *for each* $i \in I) \implies s_{|\mathbb{K}} \in \mathsf{RC}(\mathbb{K})$.

The connection policy of Example 4.9 does enjoy all the properties of Definition 3.4. Moreover, the interfaces of the four systems of Example 4.1 are all with no mixed state. Hence Theorem 5.1 guarantees that any property (among those of Definition 3.4, but lock-freedom) enjoyed by the systems is also enjoyed by their PaI multicomposition.

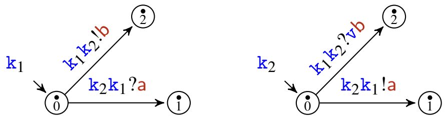
Now we provide some examples for cases in which communication properties are not preserved. First we show that all the three properties for which we have assumed the no-mixed-state condition in Theorem 5.1 would, in general, not be preserved by composition if the condition is dropped. In the counterexamples, the receiving states introduced by the gateway construction cause the breaking of the property taken into account.

Example 5.5 (No-mixed-state counterexample for deadlock-freeness and progress preservation)**.** Let us consider the two following systems S_1 and S_2 with interfaces, respectively, $\mathtt{h}_1$ and $\mathtt{h}_2$ containing mixed states.

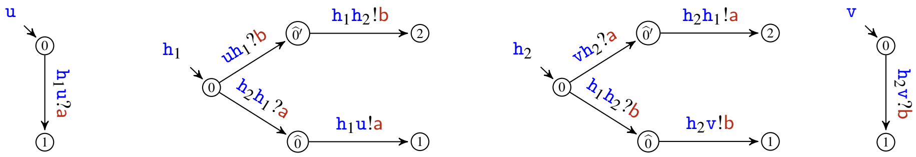


S_1 and S_2 are both deadlock free and both enjoy the progress property. There is a unique communication model for their composition: $\textsc{cm} = \{(\mathtt{h}_2,\mathtt{a},\mathtt{h}_1),(\mathtt{h}_1,\mathtt{b},\mathtt{h}_2)\}$ The unique communication policy complying with $\textsc{cm}$ is the following one.

$\mathbb{K} =$

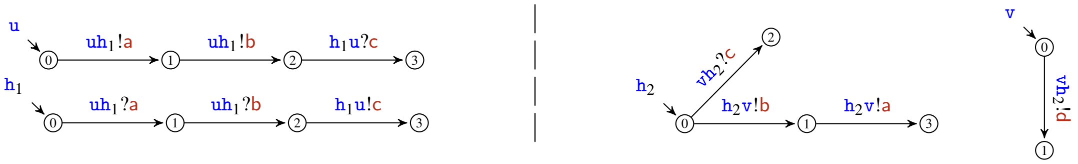


Also $\mathbb{K}$ is deadlock free and enjoys the progress property. The system $\mathscr{MC}(\{S_1,S_2\},\mathbb{K})$ is the following one.



The initial configuration is actually a deadlock, and hence the system does also not enjoy progress. ◇

Example 5.6 (No mixed-state counterexample for reception-error-freeness preservation)**.** Let us consider the two following systems S_1 and S_2 with interfaces, respectively, $\mathtt{h}_1$ and $\mathtt{h}_2$ containing mixed states.

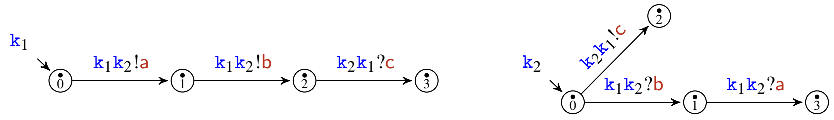


S_1 and S_2 are both reception-error free. The unique communication model for their composition is

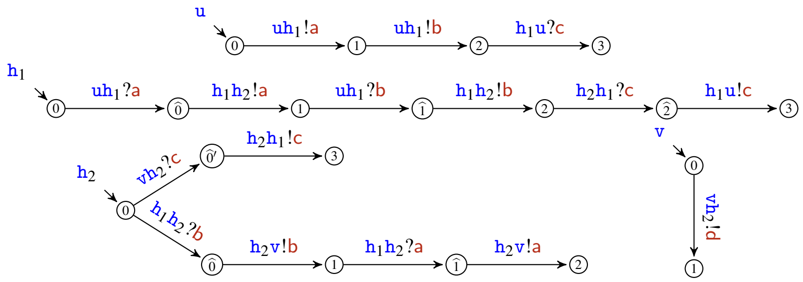
$$\textsc{cm} = \{(\mathtt{h}_1, \mathsf{a}, \mathtt{h}_2), (\mathtt{h}_1, \mathsf{b}, \mathtt{h}_2), (\mathtt{h}_2, \mathsf{c}, \mathtt{h}_1)\}$$

The unique communication policy complying with $\textsc{cm}$ is

$\mathbb{K} =$



Also $\mathbb{K}$ is reception-error free. The system $\mathscr{MC}(\{S_1, S_2\}, \mathbb{K})$ is the following one.



This communication system, however, is not reception-error free, since it is possible to reach the configuration $s = (\vec{q}, \vec{w})$ where

$$\vec{q} = (2_{\mathtt{u}}, 2_{\mathtt{h}_1}, 0_{\mathtt{h}_2}, 1_{\mathtt{v}}), \qquad w_{\mathtt{h}_1\mathtt{h}_2} = \langle \mathsf{a} \cdot \mathsf{b} \rangle, \qquad w_{\mathtt{vh}_2} = \langle \mathsf{d} \rangle, \qquad w_c = \varepsilon \ \ (\forall c \notin \{\mathtt{h}_1\mathtt{h}_2, \mathtt{vh}_2\})$$

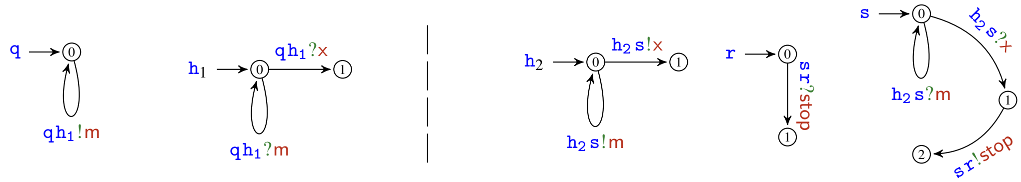
In the configuration s, the CFSM $\mathtt{h}_2$ is in a receiving state, namely 0, from which there are two transitions, namely $(0, \mathtt{vh}_2?\mathsf{c}, \hat{0}')$ and $(0, \mathtt{h}_1\mathtt{h}_2?\mathsf{b}, \hat{0})$. Moreover, the channels $\mathtt{vh}_2$ and $\mathtt{h}_1\mathtt{h}_2$ are both not empty and their first element is different from both b and c. The above configuration is hence an unspecified reception configuration. ◇

Notice that in case we dropped the requirement that $\mathbb{K}$ has to comply with a communication model, the interfaces $\mathtt{h}_1$ and $\mathtt{h}_2$ of Example 5.6 could be simplified to get the counterexample. In particular, they could have just, respectively, two and three states. The use of communication models hence limits the possibility of getting systems whose properties are not preserved by composition. This is an indication that connection models increase the possibility of getting safe compositions.

Let us now turn to the last communication property stated in Definition 3.4 which is lock-freeness. This property is also meaningful in the context of synchronous communication.

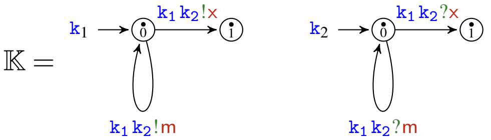
In [7, Example 6.7] a counterexample is provided, showing that in the formalism of *synchronous* CFSMs the properties of (synchronous) lock-freeness and deadlock-freeness are, in general, not preserved. As a matter of fact, lock-freeness is problematic also for the case of asynchronous communications and no mixed states, as shown in the following example, adapted from [7].

Example 5.7 (Lock-freeness is not preserved by composition)**.** Let us consider the following communicating systems S_1 and S_2.

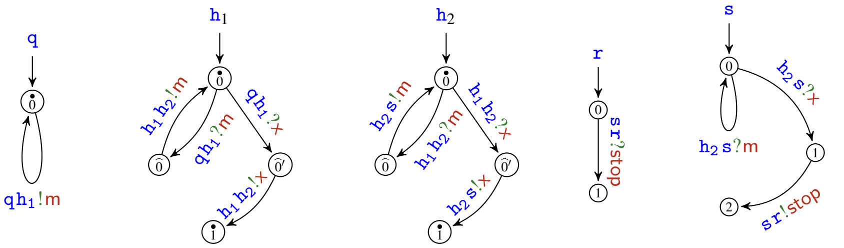


Note that both S_1 and S_2 are lock-free and their respective interfaces $\mathtt{h}_1$ and $\mathtt{h}_2$ have no mixed states.

Let us now consider the (unique) connection policy $\mathbb{K} = (M_{\mathtt{k}_i})_{i\in\{1,2\}}$ where $M_{\mathtt{k}_1} \in \mathsf{LCPS}(M_{\mathtt{h}_1}, \mathrm{CM})$ and $M_{\mathtt{k}_2} \in \mathsf{LCPS}(M_{\mathtt{h}_2}, \mathrm{CM})$ with connection model $\mathrm{CM} = \{(\mathtt{h}_1, \mathsf{x}, \mathtt{h}_2), (\mathtt{h}_1, \mathsf{m}, \mathtt{h}_2)\}$.



It is easy to see that $\mathbb{K}$ is lock-free. The multicomposition $\mathscr{MC}(\{S_i\}_{i\in\{1,2\}}, \mathbb{K})$ is the following communicating system:



The initial configuration s_0 of $\mathscr{MC}(\{S_i\}_{i\in\{1,2\}}, \mathbb{K})$ is an $\mathtt{r}$-lock, since the transition $\mathtt{q}\,\mathtt{h}_1?\mathsf{x}$ of $\mathtt{h}_1$ can never be fired, so implying, in turn, that also $\mathtt{h}_1\,\mathtt{h}_2!\mathsf{x}$ of $\mathtt{h}_1$, $\mathtt{h}_1\,\mathtt{h}_2?\mathsf{x}$ of $\mathtt{h}_2$, $\mathtt{h}_2\,\mathtt{s}!\mathsf{x}$ of $\mathtt{h}_2$, $\mathtt{h}_2\,\mathtt{s}?\mathsf{x}$ of $\mathtt{s}$ and $\mathtt{s}\,\mathtt{r}!\mathsf{stop}$ of $\mathtt{s}$ can never be fired. Hence, no transition sequence out of s_0 will ever involve the participant $\mathtt{r}$. Thus $\mathscr{MC}(\{S_i\}_{i\in\{1,2\}}, \mathbb{K})$ is not lock-free. ◇

6 Conclusions

The necessity of supporting the modular development of concurrent/distributed systems, as well as the need to extend/modify/adapt/upgrade them, urged the investigation of composition methods. Focusing on such investigations in the setting of abstract formalisms for the description and verification of systems enables to get general and formal guarantees of relevant features of the composition methods.

An investigation of composition in a formalism for choreographic programming was carried out in [24]. In [22] a modular technique was developed for the verification of aspect-oriented programs expressed as state machines. Team Automata is another formalism in which compositionality issues have been addressed [9, 8], as well as in assembly theories considered in [19]. Composition for protocols described via a process algebra has been investigated in [10]. In [13, 25] a technique for modular design in the setting of reactive programming is proposed. A possible approach to composition for a MultiParty Session Type (MPST) formalism is developed in [26]. The mentioned papers provide just a glimpse of the variety of approaches to system composition in the literature.

Papers dealing with the (binary) composition of systems on the basis of the *participants-as-interfaces* (PaI) approach have been pointed out already in Section 1 and the idea of PaI for multicomposition of systems has been explained in Section 2. In the present paper we study the PaI approach to multicomposition for systems of asynchronously communicating finite state machines (CFSMs). We show that (under mild assumptions) important communication properties relevant in the context of asynchronous communication, like freeness of orphan messages and unspecified receptions, are preserved by composition (a feature dubbed *safety* in [3]). For this we assume that for each single system one participant is chosen as an interface. A key role in our work, inspired by [3], is played by *connection policies*, which are CFSM systems which determine the ways how interfaces can interact when they are replaced by gateways (forwarders) in system compositions.

For an "unstructured" formalism like CFSM, the natural generalisation from multicomposition with single interfaces to multicomposition with multiple interfaces (per system) is not trouble-free, as discussed in [1, Sect.6] for binary composition. This is mainly due to the possible indirect interactions which could occur among the interfaces inside the single systems. In more structured formalisms, however, such possible interactions can be controlled. This is the case, for instance, in MPST formalisms. In fact, in [17] the authors devise a direct composition mechanism without using gateways for MPST systems. Such a mechanism allows for the presence of multiple interfaces thanks to an hybridisation with local and external information of the standard notion of global type. A combination of global and local constructs in order to get flexible specifications (uniformly describing both the internal and the interface behavior of systems) is also present in [12].

There are several directions to be pursued in future work starting from our results. On the first place, we want to generalise the notion of connection policy such that PaI multicomposition could actually be obtained by replacing interfaces by gateways which, instead of interacting directly with each other, can interact through an "interfacing infrastructure" represented via a system of CFSMs. Such a generalisation would be equivalent to multicomposition where exactly one system is enabled to have multiple interfaces. Let us consider a possible application of the above idea. In Example 4.1, in the resulting composed system, both participants $\mathtt{p}$ and $\mathtt{q}$ do emit a react message. It would be more natural to have only one of them produce such a message, e.g., to have $\mathtt{p}$ be the sole sensor registering reactions which then passes that information to both $\mathtt{r}$ and $\mathtt{s}$. This would not be possible by our composition mechanisms and we cannot but make the best of the fact that we are dealing with two sensors. One could think, instead, about using an "interfacing infrastructure" containing some further participant enabling to ignore the messages from one sensor and properly duplicating the messages from the other.

It is worth noticing how, in Examples 5.5, 5.6 and 5.7, the interfaces of the systems we compose do have unreachable states. It is hence natural to wonder whether it is the presence of unreachable states in interfaces that entails the possibility of getting counterexamples for the properties taken into account.

We are also planning to consider further communication properties, like strong lock-freeness (any participant can eventually progress in any continuation of any reachable configuration), as well as to investigate conditions to get lock-freeness preservation, not guaranteed yet.

Unlike the present paper, in [1] safeness is ensured for the binary case by assuming compatibility of interfaces and an extra condition (called ?!-determinism) on them. We are currently considering a generalisation of the binary compatibility relation. Such generalisation should imply relevant communication properties for the communication policy it depends on.

We are planning also to identify some conditions ensuring Theorem 5.1 to hold for any communication property $\mathscr{P}$ satisfying them.

Finally, we could consider "partial" gateways, where only some communications of an interface are interpreted as communications with the environment. Such an idea was actually implemented in [2] in a MPTS setting for a restricted client-multiserver composition with synchronous communications.

Acknowledgements We warmly thank the ICE'24 reviewers for their careful reading, their thoughtful comments/suggestions and the helpful discussion on the forum. We also thank Emilio Tuosto for his nice tikz style for automata.

References

[1] Franco Barbanera, Ugo de'Liguoro & Rolf Hennicker (2019): *Connecting open systems of communicating finite state machines*. J. Log. Algebraic Methods Program. 109, article 100476, doi:10.1016/J.JLAMP.2019.07.004.

[2] Franco Barbanera, Mariangiola Dezani-Ciancaglini & Ugo de'Liguoro (2022): *Open compliance in multiparty sessions*. In S. Lizeth Tapia Tarifa & José Proença, editors: Proc. FACS 2022, LNCS 13712, Springer, pp. 222–243, doi:10.1007/978-3-031-20872-0_13. Extended version at `http://www.di.unito.it/~dezani/papers/bd23b.pdf`.

[3] Franco Barbanera, Mariangiola Dezani-Ciancaglini, Lorenzo Gheri & Nobuko Yoshida (2023): *Multicompatibility for Multiparty-Session Composition*. In Santiago Escobar & Vasco T. Vasconcelos, editors: Proc. PPDP 2023, ACM, pp. 2:1–2:15, doi:10.1145/3610612.3610614.

[4] Franco Barbanera, Mariangiola Dezani-Ciancaglini, Ivan Lanese & Emilio Tuosto (2021): *Composition and decomposition of multiparty sessions*. J. Log. Algebraic Methods Program. 119, article 100620, doi:10.1016/j.jlamp.2020.100620.

[5] Franco Barbanera, Ivan Lanese & Emilio Tuosto (2020): *Composing communicating systems, synchronously*. In Tiziana Margaria & Bernhard Steffen, editors: Proc. ISoLA 2020, LNCS 12476, Springer, pp. 39–59, doi:10.1007/978-3-030-61362-4_3.

[6] Franco Barbanera, Ivan Lanese & Emilio Tuosto (2022): *On composing communicating systems*. In Clément Aubert, Cinzia Di Giusto, Larisa Safina & Alceste Scalas, editors: Proc. ICE 2022, EPTCS 365, Open Publishing Association, pp. 53–68, doi:10.4204/EPTCS.365.4.

[7] Franco Barbanera, Ivan Lanese & Emilio Tuosto (2023): *Composition of synchronous communicating systems*. J. Log. Algebraic Methods Program. 135, article 100890, doi:10.1016/J.JLAMP.2023.100890.

[8] Maurice H. ter Beek, Rolf Hennicker & Jetty Kleijn (2020): *Compositionality of Safe Communication in Systems of Team Automata*. In Violet Ka I Pun, Volker Stolz & Adenilso Simão, editors: Proc. ICTAC 2020, LNCS 12545, Springer, pp. 200–220, doi:10.1007/978-3-030-64276-1_11.

[9] Maurice H. ter Beek & Jetty Kleijn (2003): *Team Automata Satisfying Compositionality*. In Keijiro Araki, Stefania Gnesi & Dino Mandrioli, editors: Proc. FME 2003, LNCS 2805, Springer, pp. 381–400, doi:10.1007/978-3-540-45236-2_22.

[10] Laura Bocchi, Dominic Orchard & A. Laura Voinea (2023): *A Theory of Composing Protocols*. Art Sci. Eng. Program. 7(2), doi:10.22152/PROGRAMMING-JOURNAL.ORG/2023/7/6. Article 6.

[11] Daniel Brand & Pitro Zafiropulo (1983): *On Communicating Finite-State Machines*. J. ACM 30(2), pp. 323–342, doi:10.1145/322374.322380.

[12] Luís Caires & Hugo Torres Vieira (2010): *Conversation types*. Theor. Comput. Sci. 411(51-52), pp. 4399–4440, doi:10.1016/J.TCS.2010.09.010.

[13] Marco Carbone, Fabrizio Montesi & Hugo Torres Vieira (2018): *Choreographies for Reactive Programming*. CoRR abs/1801.08107. Available at `http://arxiv.org/abs/1801.08107`.

[14] Gérard Cécé & Alain Finkel (2005): *Verification of programs with half-duplex communication*. Inf. Comput. 202(2), pp. 166–190, doi:10.1016/j.ic.2005.05.006.

[15] Lorenzo Clemente, Frédéric Herbreteau & Grégoire Sutre (2014): *Decidable Topologies for Communicating Automata with FIFO and Bag Channels*. In Paolo Baldan & Daniele Gorla, editors: Proc. CONCUR 2014, LNCS 8704, Springer, pp. 281–296, doi:10.1007/978-3-662-44584-6_20.

[16] Pierre-Malo Deniélou & Nobuko Yoshida (2012): *Multiparty Session Types Meet Communicating Automata*. In Helmut Seidl, editor: Proc. ESOP 2012, pp. 194–213, doi:10.1007/978-3-642-28869-2_10.

[17] Lorenzo Gheri & Nobuko Yoshida (2023): *Hybrid Multiparty Session Types: Compositionality for Protocol Specification through Endpoint Projection*. Proc. ACM Program. Lang. 7(OOPSLA1), pp. 112–142, doi:10.1145/3586031.

[18] Rolf Hennicker & Michel Bidoit (2018): *Compatibility Properties of Synchronously and Asynchronously Communicating Components*. *Log. Meth. in Comp. Sci.* 14(1), pp. 1–31, doi:10.23638/LMCS-14(1:1)2018.

[19] Rolf Hennicker & Alexander Knapp (2015): *Moving from interface theories to assembly theories*. *Acta Informatica* 52(2-3), pp. 235–268, doi:10.1007/S00236-015-0220-7.

[20] Kohei Honda, Nobuko Yoshida & Marco Carbone (2008): *Multiparty asynchronous session types*. In George C. Necula & Philip Wadler, editors: *Proc. POPL 2008*, ACM, pp. 273–284, doi:10.1145/1328438.1328472.

[21] Kohei Honda, Nobuko Yoshida & Marco Carbone (2016): *Multiparty asynchronous session types*. *J. ACM* 63(1), pp. 9:1–9:67, doi:10.1145/2827695.

[22] Shriram Krishnamurthi, Kathi Fisler & Michael Greenberg (2004): *Verifying aspect advice modularly*. In Richard N. Taylor & Matthew B. Dwyer, editors: *Proc. SIGSOFT 2004*, ACM, pp. 137–146, doi:10.1145/1029894.1029916.

[23] Julien Lange, Emilio Tuosto & Nobuko Yoshida (2015): *From Communicating Machines to Graphical Choreographies*. In Sriram K. Rajamani & David Walker, editors: *Proc. POPL 2015*, ACM, pp. 221–232, doi:10.1145/2676726.2676964.

[24] Fabrizio Montesi & Nobuko Yoshida (2013): *Compositional Choreographies*. In Pedro R. D'Argenio & Hernán C. Melgratti, editors: *Proc. CONCUR 2013*, *LNCS* 8052, Springer, pp. 425–439, doi:10.1007/978-3-642-40184-8_30.

[25] Zorica Savanovic, Letterio Galletta & Hugo Torres Vieira (2020): *A type language for message passing component-based systems*. In Julien Lange, Anastasia Mavridou, Larisa Safina & Alceste Scalas, editors: *Proc. ICE 2020*, *EPTCS* 324, pp. 3–24, doi:10.4204/EPTCS.324.3.

[26] Claude Stolze, Marino Miculan & Pietro Di Gianantonio (2023): *Composable partial multiparty session types for open systems*. *Softw. Syst. Model.* 22(2), pp. 473–494, doi:10.1007/S10270-022-01040-X.

[27] Emilio Tuosto & Roberto Guanciale (2018): *Semantics of global view of choreographies*. *J. Log. Algebr. Meth. Program.* 95, pp. 17–40, doi:10.1016/j.jlamp.2017.11.002.

A Proof of Theorem 5.1 (Preservation of Communication Properties)

In this appendix we prove Theorem 5.1. For convenience we repeat below the definition of configuration projection Definition 5.3 (now with number Definition A.1) and Proposition 5.4 (now with number A.5). In the following we make the following assumptions (if not otherwise specified).

General assumptions:
We assume given a system S obtained via multicomposition:

$$S = (M_{\mathtt{p}})_{\mathtt{p}\in\mathbf{P}} = \mathscr{MC}(\{S_i\}_{i\in I}, \mathbb{K})$$

where $\{S_i\}_{i\in I}$ is a set of CSs, composable with respect to the set of interfaces $H = \{\mathtt{h}_i\}_{i\in I}$, and where $S_i = (M^i_{\mathtt{p}})_{\mathtt{p}\in\mathbf{P}_i}$ for each $i \in I$. Notice that, for each $i \in I$ and $\mathtt{p} \in \mathbf{P}_i \setminus \{\mathtt{h}_i\}$ we have $M^i_{\mathtt{p}} = M_{\mathtt{p}}$. Moreover, we assume that $\mathbb{K} = (M_{\mathtt{k}_i})_{i\in I}$ is a connection policy (complying with a given connection model CM for H though this is only of methodological relevance and irrelevant for the proofs).

Notations:
The channels of S_i are $C_i = \{\mathtt{pq} \mid \mathtt{p},\mathtt{q} \in \mathbf{P}_i, \mathtt{p} \neq \mathtt{q}\}$.
The channels of S are $C = \bigcup_{i\in I} C_i \cup \{\mathtt{h}_i\mathtt{h}_j\}_{i,j\in I, i\neq j}$.
The set of transitions of $M_{\mathtt{p}}$ in S is denoted by $\delta_{\mathtt{p}}$.[4]
The set of transitions of $M^i_{\mathtt{p}}$ in S_i will be denoted by $\delta^i_{\mathtt{p}}$ where $\delta^i_{\mathtt{p}} = \delta_{\mathtt{p}}$ for each $\mathtt{p} \in \mathbf{P}_i \setminus \{\mathtt{h}_i\}$.
The set of transitions of $M_{\mathtt{k}_i}$ will be denoted by $\delta^{\mathbb{K}}_{\mathtt{k}_i}$.

A.1 Technical notions and results

Definition A.1 (Configuration projection)**.** *Let $s = (\vec{q}, \vec{w}) \in \mathsf{RC}(S)$ where $\vec{q} = (q_{\mathtt{p}})_{\mathtt{p}\in\mathbf{P}}$ and $\vec{w} = (w_{\mathtt{pq}})_{\mathtt{pq}\in C}$. For each $i \in I$, the projection $s_{|_{\mathbf{i}}}$ of s to S_i is defined by*

$$s_{|_{\mathbf{i}}} = (\vec{q}_{|_{\mathbf{i}}}, \vec{w}_{|_{\mathbf{i}}})$$

where $\vec{q}_{|_{\mathbf{i}}} = (q_{\mathtt{p}})_{\mathtt{p}\in\mathbf{P}_i}$ and $\vec{w}_{|_{\mathbf{i}}} = (w_{\mathtt{pq}})_{\mathtt{pq}\in C_i}$.
The projection $s_{|_{\mathbb{K}}}$ of $s = (\vec{q}, \vec{w})$ to $\mathbb{K}$ is defined if $q_{\mathtt{h}_i} \notin \widehat{Q}_{\mathtt{h}_i}$ for each $i \in I$ and then

$$s_{|_{\mathbb{K}}} = (\vec{q}_{|_{\mathbb{K}}}, \vec{w}_{|_{\mathbb{K}}})$$

where $\vec{q}_{|_{\mathbb{K}}} = (p_{\mathtt{k}_i})_{i\in I}$ is such that, for each $i \in I$, $p_{\mathtt{k}_i} = \dot{q}_{\mathtt{h}_i}$
and where $\vec{w}_{|_{\mathbb{K}}} = (w'_{\mathtt{pq}})_{\mathtt{p},\mathtt{q}\in\{\mathtt{k}_i\}_{i\in I}, \mathtt{p}\neq\mathtt{q}}$ is such that, for each pair $i, j \in I$ with $i \neq j$, $w'_{\mathtt{k}_i\mathtt{k}_j} = w_{\mathtt{h}_i\mathtt{h}_j}$.

The following fact easily descends from how gateways are built (Definition 4.11). In particular from the fact that the gateway transformation of a machine M does insert an intermediate state between any pair of states of M connected by a transition. By definition, the intermediate state possesses exactly one incoming transition and one outgoing transition.

Fact A.2.
Let $s = (\vec{q}, \vec{w}) \in \mathsf{RC}(S)$. For each $i \in I$, by setting $\mathtt{h} = \mathtt{h}_i$ and $\mathtt{k} = \mathtt{k}_i$, the following holds, where $M_{\mathtt{h}} \hookleftarrow M_{\mathtt{k}} = (Q_{\mathtt{h}} \cup \widehat{Q_{\mathtt{h}}}, \_, \_, \delta_{\mathtt{h}})$.

1. *If $q_{\mathtt{h}} \in \widehat{Q_{\mathtt{h}}}$ then $q_{\mathtt{h}}$ is not final and there exists a unique transition $(q_{\mathtt{h}}, \_, \_) \in \delta_{\mathtt{h}}$. Moreover, such a transition is of the form $(q_{\mathtt{h}}, \mathtt{hs}!a, q')$ with $q' \notin \widehat{Q_{\mathtt{h}}}$.*

[4] Note that, for $\mathtt{h} \in \{\mathtt{h}_i\}_{i\in I}$, we denote by $\delta_{\mathtt{h}}$ for what is written $\widehat{\delta}$ in the definition of $M_{\mathtt{h}} \hookleftarrow M_{\mathtt{k}}$ (see Definition 4.11).

2. *If $q_{\mathtt{h}} \notin \widehat{Q_{\mathtt{h}}}$ then either $q_{\mathtt{h}}$ is final, or any transition $(q_{\mathtt{h}},\_,\_) \in \delta_{\mathtt{h}}$ is an input one, that is of the form $(q_{\mathtt{h}},\mathtt{sh}?a,q'_{\mathtt{h}})$ with $q'_{\mathtt{h}} \in \widehat{Q_{\mathtt{h}}}$.*
3. *If $q_{\mathtt{h}} \notin \widehat{Q_{\mathtt{h}}}$ then*
   - *a) If $(q_{\mathtt{h}},\mathtt{h}_j\mathtt{h}?a,q'_{\mathtt{h}}) \in \delta_{\mathtt{h}}$ for a $j \in I$, then there exists $(q'_{\mathtt{h}},\mathtt{hs}!a,q''_{\mathtt{h}}) \in \delta_{\mathtt{h}}$ with $\mathtt{s} \in \mathbf{P}_i$ (and hence, for each $l \in I$, $\mathtt{s} \neq \mathtt{h}_l$) such that $(q_{\mathtt{h}},\mathtt{hs}!a,q''_{\mathtt{h}}) \in \delta^i_{\mathtt{h}}$.*
   - *b) If $(q_{\mathtt{h}},\mathtt{sh}?a,q'_{\mathtt{h}}) \in \delta_{\mathtt{h}}$ with $\mathtt{s} \in \mathbf{P}_i$ (and hence, for each $j \in I$, $\mathtt{s} \neq \mathtt{h}_j$) then there exists $(q'_{\mathtt{h}},\mathtt{hh}_j!a,q''_{\mathtt{h}}) \in \delta_{\mathtt{h}}$, for a $j \in I$, such that $(q_{\mathtt{h}},\mathtt{sh}?a,q''_{\mathtt{h}}) \in \delta^i_{\mathtt{h}}$.*

Lemma A.3. *Let $S = \mathscr{MC}(\{S_i\}_{i\in I},\mathbb{K})$ and $\mathtt{h} \in \{\mathtt{h}_i\}_{i\in I}$.*

i) Let $s,s',s'' \in \mathsf{RC}(S)$ such that

$$s \xrightarrow{\mathtt{rh}?\mathtt{a}} s' \xrightarrow{l} s'' \text{ where } l \text{ is not of the form } \mathtt{h}\_!\_.$$

Then there exists $s''' \in \mathsf{RC}(S)$ such that $s \xrightarrow{l} s''' \xrightarrow{\mathtt{rh}?\mathtt{a}} s''$.

ii) For $j,k \geq 0$, let $s_j \in \mathsf{RC}(S)$ and $s_{j+k+1} \in \mathsf{RC}(S)$ be reachable from s_j by a sequence of transitions of the form $s_j \xrightarrow{\mathtt{rh}?\mathtt{a}} s_{j+1} \xrightarrow{l_1} \ldots s_{j+k} \xrightarrow{l_k} s_{j+k+1}$ where, for $x = 1,\ldots,k$, l_x is not of the form $\mathtt{h}\_!\_$.
Then there exists a sequence of transitions of the form

$$s_j \xrightarrow{l_1} s'_{j+1} \ldots \xrightarrow{l_k} s'_{j+k} \xrightarrow{\mathtt{rh}?\mathtt{a}} s_{j+k+1}.$$

iii) Let $s \in \mathsf{RC}(S)$ be reachable from s_0 by a sequence of transitions of the form

$$s_0 \to s_1 \ldots \to s_{n-2} \to s_{n-1} \xrightarrow{\mathtt{hs}!\mathtt{a}} s_n = s.$$

Then $n \geq 2$ and there exists a sequence of transitions of the form

$$s_0 \to s'_1 \ldots \to s'_{n-2} \xrightarrow{\mathtt{rh}?\mathtt{a}} s_{n-1} \xrightarrow{\mathtt{hs}!\mathtt{a}} s_n = s.$$

Proof. (i) Let $s = (\vec{q},\vec{w}), s' = (\vec{q'},\vec{w'})$ and $s'' = (\vec{q''},\vec{w''})$. For all $\mathtt{p} \in \mathbf{P} \setminus \{\mathtt{h}\}$ we have $q'_{\mathtt{p}} = q_{\mathtt{p}}$. Since l is not of the form $\mathtt{h}\_!\_$ by assumption and l is also not of the form $\_\mathtt{h}?\_$ by gateway construction, we have $q''_{\mathtt{h}} = q'_{\mathtt{h}}$. Then we set $q'''_{\mathtt{h}} = q_{\mathtt{h}}$ and $q'''_{\mathtt{p}} = q''_{\mathtt{p}}$ for all $\mathtt{p} \in \mathbf{P} \setminus \{\mathtt{h}\}$.

Concerning the channels, we know that $w_{\mathtt{rh}} = \mathtt{a} \cdot w'_{\mathtt{rh}}$ and $w'_{\mathtt{pv}} = w_{\mathtt{pv}}$ for all $\mathtt{pv} \neq \mathtt{rh}$. Now we set $w'''_{\mathtt{pv}} = w''_{\mathtt{pv}}$ for all $\mathtt{pv} \neq \mathtt{rh}$. For defining $w'''_{\mathtt{rh}}$ we consider three cases for l:

- $\diamond$ $l = \mathtt{rh}?\mathtt{b}$ for some $\mathtt{b}$.

  As already said above, this case is not possible by construction of gateways.

- $\diamond$ $l \neq \mathtt{rh}?\mathtt{b}$ and $l \neq \mathtt{rh}!\mathtt{b}$ for any $\mathtt{b}$.

  Then $w''_{\mathtt{rh}} = w'_{\mathtt{rh}}$. We set $w'''_{\mathtt{rh}} = w_{\mathtt{rh}}$ and $s''' = (\vec{q'''},\vec{w'''})$. Thus $s \xrightarrow{l} s''' \xrightarrow{\mathtt{rh}?\mathtt{a}} s''$.

- $\diamond$ $l = \mathtt{rh}!\mathtt{b}$ for some $\mathtt{b}$.

  Then $w''_{\mathtt{rh}} = w'_{\mathtt{rh}} \cdot \mathtt{b}$. We set $w'''_{\mathtt{rh}} = w_{\mathtt{rh}} \cdot \mathtt{b} = \mathtt{a} \cdot w'_{\mathtt{rh}} \cdot \mathtt{b}$ and $s''' = (\vec{q'''},\vec{w'''})$. Thus $s \xrightarrow{l} s''' \xrightarrow{\mathtt{rh}?\mathtt{a}} s''$.

(ii) The proof is done by induction on k.

$k = 0$. Then we take $s'_{j+k} = s_j$ and the statement is trivial.

$k \mapsto k+1$. Let $s_j \xrightarrow{\mathtt{rh}?\mathtt{a}} s_{j+1} \xrightarrow{l_1} s_{j+2} \ldots \xrightarrow{l_k} s_{j+k+1} \xrightarrow{l_{k+1}} s_{j+k+2}$. Then, by part (i) of the lemma, there exists s'_{j+1} such that $s_j \xrightarrow{l_1} s'_{j+1} \xrightarrow{\mathtt{rh}?\mathtt{a}} s_{j+2}$. By induction hypothesis, there exists a sequence of transitions $s'_{j+1} \xrightarrow{l_2} \ldots \xrightarrow{l_{k+1}} s'_{j+k+1} \xrightarrow{\mathtt{rh}?\mathtt{a}} s_{j+k+2}$. Thus $s_j \xrightarrow{l_1} s'_{j+1} \xrightarrow{l_2} \ldots \xrightarrow{l_{k+1}} s'_{j+k+1} \xrightarrow{\mathtt{rh}?\mathtt{a}} s_{j+k+2}$.

(iii) Let $s_0 \to s_1 \ldots \to s_{n-2} \to s_{n-1} \xrightarrow{\mathtt{hs}!\mathtt{a}} s_n = s$ with $s = (\vec{q},\vec{w})$. Due to the construction of gateways $q_{(n-1)_{\mathtt{h}_i}} \in \widehat{Q_{\mathtt{h}_i}}$ for each $i \in I$ and, since $q_{0_{\mathtt{h}_i}} \notin \widehat{Q_{\mathtt{h}_i}}$ for each $i \in I$, there must be a transition $s_j \xrightarrow{\mathtt{rh}?\mathtt{a}} s_{j+1}$ for some $0 \leq j \leq n-2$. In particular, $n \geq 2$ must hold. Now we take the largest j with $0 \leq j \leq n-2$

such that $s_j \xrightarrow{\mathtt{rh?a}} s_{j+1} \xrightarrow{l_1} \dots s_{j+k} \xrightarrow{l_k} s_{n-1}$ where, for each $x = 1,\dots,k$, l_x is not of the form $\mathtt{h}\_!\_$. By part (ii) of the lemma, there exists $s_j \xrightarrow{l_1} s'_{j+1} \dots \xrightarrow{l_k} s'_{j+k} \xrightarrow{\mathtt{rh?a}} s_{n-1}$. Thus we obtain a sequence $s_0 \to \dots \to s_j \xrightarrow{l_1} s'_{j+1} \dots \xrightarrow{l_k} s'_{n-2} \xrightarrow{\mathtt{rh?a}} s_{n-1} \xrightarrow{\mathtt{hs!a}} s_n = s$.

□

Lemma A.4. *Let $s \in \mathsf{RC}(S)$ be a reachable configuration of $S = \mathscr{MC}(\{S_i\}_{i\in I}, \mathbb{K})$. Then, for each $i \in I$,*

i) $s_{0|_{\mathtt{i}}} \in \mathsf{RC}(S_i)$.

ii) Let $s \xrightarrow{l} s'$. If l is neither of the form $\_\mathtt{h}_i?\_$ nor of the form $\mathtt{h}_i\_!\_$, then either $s_{|_{\mathtt{i}}} \xrightarrow{l} s'_{|_{\mathtt{i}}}$ or $s_{|_{\mathtt{i}}} = s'_{|_{\mathtt{i}}}$.

iii) $s \xrightarrow{\mathtt{rh}_i\mathtt{?a}} s' \xrightarrow{\mathtt{h}_i\mathtt{s!a}} s''$ implies $s_{|_{\mathtt{i}}} \xrightarrow{l} s''_{|_{\mathtt{i}}}$, where $l = \mathtt{rh}_i\mathtt{?a}$ if $\mathtt{r} \in \mathbf{P}_i$ and $l = \mathtt{h}_i\mathtt{s!a}$ if $\mathtt{s} \in \mathbf{P}_i$.

Proof. (i) is trivially true.
(ii) Let $s = (\vec{q}, \vec{w})$ and $s' = (\vec{q'}, \vec{w'})$. We consider two cases for l:

- $\diamond$ l is of the form $\mathtt{rs?a}$ or $\mathtt{rs!a}$ with $\mathtt{r},\mathtt{s} \in \mathbf{P}_i$.
  Since l is neither of the form $\_\mathtt{h}_i?\_$ nor of the form $\mathtt{h}_i\_!\_$ it holds that $q_{\mathtt{h}_i}, q'_{\mathtt{h}_i} \notin \widehat{Q_{\mathtt{h}_i}}$ (by construction of gateways) and, in particular, $q_{\mathtt{h}_i} = q'_{\mathtt{h}_i}$. Then, by definition of projection, we get $s_{|_{\mathtt{i}}} \xrightarrow{l} s'_{|_{\mathtt{i}}}$.
- $\diamond$ l is of the form $\mathtt{rs?a}$ or $\mathtt{rs!a}$ with $\mathtt{r},\mathtt{s} \in \mathbf{P}_j \cup \{\mathtt{h}_k\}_{k\in I}$, $j \neq i$. Since l is neither of the form $\_\mathtt{h}_i?\_$ nor of the form $\mathtt{h}_i\_!\_$ it holds that $q_{\mathtt{v}} = q'_{\mathtt{v}}$ for all $\mathtt{v} \in \mathbf{P}_i$ and $w_{\mathtt{uv}} = w'_{\mathtt{uv}}$ for all $\mathtt{u},\mathtt{v} \in \mathbf{P}_i$. Hence, $s_{|_{\mathtt{i}}} = s'_{|_{\mathtt{i}}}$.

(iii) follows from the construction of gateways. In particular, either $\{\mathtt{r},\mathtt{s}\} \cap \mathbf{P}_i = \mathtt{r}$ or $\{\mathtt{r},\mathtt{s}\} \cap \mathbf{P}_i = \mathtt{s}$. □

Notice that $s_{|_{\mathtt{i}}}$ is not necessarily a configuration of S_i, because of the possible presence of the additional states introduced by the gateways construction. If, however, reachable configurations of the connected system $S = \mathscr{MC}(\{S_i\}_{i\in I}, \mathbb{K})$ do not involve intermediate states of gateways, then, by projection, we get reachable configurations of S_i and $\mathbb{K}$.

Proposition A.5 (On reachability of projections)**.** *Let $s = (\vec{q}, \vec{w}) \in \mathsf{RC}(S)$.*

i) For each $i \in I$, $(q_{\mathtt{h}_i} \notin \widehat{Q_{\mathtt{h}_i}} \implies s_{|_{\mathtt{i}}} \in \mathsf{RC}(S_i))$;

ii) $(q_{\mathtt{h}_i} \notin \widehat{Q_{\mathtt{h}_i}}$ for each $i \in I) \implies s_{|_{\mathbb{K}}} \in \mathsf{RC}(\mathbb{K})$.

Proof. (i) Let $s = (\vec{q}, \vec{w}) \in \mathsf{RC}(S)$ and $i \in I$ such that $q_{\mathtt{h}_i} \notin \widehat{Q_{\mathtt{h}_i}}$. We prove $s_{|_{\mathtt{i}}} \in \mathsf{RC}(S_i)$ by (well-founded) induction on the length n of the transition sequence to reach s from the initial state s_0.

Case $n = 0$. Then $s_{|_{\mathtt{i}}} = s_{0|_{\mathtt{i}}} \in \mathsf{RC}(S_i)$ by Lemma A.4(i).

Case $n > 0$. Then there exists $s_x = (\vec{q}_x, \vec{w}_x) \in \mathsf{RC}(S)$ and an action l_x over $\mathbf{P}$ such that $s_x \xrightarrow{l_x} s$ and s_x is reachable from s_0 in $n-1$ steps. We consider the following cases:

- $\diamond$ l_x is neither of the form $\_\mathtt{h}_i?\_$ nor of the form $\mathtt{h}_i\_!\_$.
  Then $q_{x\mathtt{h}_i} = q_{\mathtt{h}_i} \notin \widehat{Q_{\mathtt{h}_i}}$. Moreover, by Lemma A.4(ii), either $s_{x|_{\mathtt{i}}} \xrightarrow{l_x} s_{|_{\mathtt{i}}}$ or $s_{x|_{\mathtt{i}}} = s_{|_{\mathtt{i}}}$.
  Since $q_{x\mathtt{h}_i} \notin \widehat{Q_{\mathtt{h}_i}}$ we can apply the induction hypothesis for s_x and obtain $s_{x|_{\mathtt{i}}} \in \mathsf{RC}(S_i)$.
  Hence $s_{|_{\mathtt{i}}} \in \mathsf{RC}(S_i)$.
- $\diamond$ l_x is of the form $\_\mathtt{h}_i?\_$.
  This is not possible since otherwise, by definition of gateways, $q_{\mathtt{h}_i} \in \widehat{Q_{\mathtt{h}_i}}$ which is excluded.

$\diamond$ l_x is of the form $\mathtt{h}_i\_!\_$.

More explicitly, let $s_x \xrightarrow{\mathtt{h}_i\mathtt{s}!\mathtt{a}} s$. Then, by Lemma A.3(iii), there exist transitions $s_y \xrightarrow{\mathtt{rh}_i?\mathtt{a}} s_x \xrightarrow{\mathtt{h}_i\mathtt{s}!\mathtt{a}} s$ such that $s_y = (\vec{q}_y, \vec{w}_y) \in \mathsf{RC}(S)$ is reachable from s_0 in $n-2$ steps. By definition of gateways, $q_{y_{\mathtt{h}_i}} \notin \widehat{Q_{\mathtt{h}_i}}$. Hence, we can apply the induction hypothesis for s_y and obtain $s_{y_{|_\mathbf{i}}} \in \mathsf{RC}(S_i)$. Moreover, by Lemma A.4(iii) we get a transition $s_{y_{|_\mathbf{i}}} \xrightarrow{l} s_{|_\mathbf{i}}$. Hence $s_{|_\mathbf{i}} \in \mathsf{RC}(S_i)$.

(ii). Let $s = (\vec{q}, \vec{w}) \in \mathsf{RC}(S)$ such that $q_{\mathtt{h}_i} \notin \widehat{Q_{\mathtt{h}_i}}$ for all $i \in I$. We prove $s_{|_\mathbb{K}} \in \mathsf{RC}(\mathbb{K})$ by (well-founded) induction on the length n of the transition sequence to reach s from the initial state s_0.

Case $n = 0$. In this case $s_0 = ((q_{0_\mathtt{p}})_{\mathtt{p}\in\mathbf{P}}, \vec{\varepsilon})$. Then, by definitions of connection policy and projection (Definition A.1 and Definition 4.8), $s_{0_{|_\mathbb{K}}} = ((p_{\mathtt{k}_i})_{i\in I}, (w'_{\mathtt{pq}})_{\mathtt{p},\mathtt{q}\in\{\mathtt{k}_i\}_{i\in I}, \mathtt{p}\neq\mathtt{q}})$ where $p_{\mathtt{k}_i} = \dot{q}_{0_{\mathtt{h}_i}}$ (since $q_{0_{\mathtt{h}_i}} \notin \widehat{Q}_{\mathtt{h}_i}$ for all $i \in I$) and $w'_{\mathtt{k}_i\mathtt{k}_j} = \varepsilon$ for each $i, j \in I$ with $i \neq j$. Obviously, $s_{0_{|_\mathbb{K}}}$ is the initial configuration of $\mathbb{K}$ and thus $s_{0_{|_\mathbb{K}}} \in \mathsf{RC}(\mathbb{K})$.

Case $n > 0$. Then there exists $s_x = (\vec{q}_x, \vec{w}_x) \in \mathsf{RC}(S)$ and an action l_x over $\mathbf{P}$ such that $s_x \xrightarrow{l_x} s$ and s_x is reachable from s_0 in $n-1$ steps. We consider the following cases:

$s_x \xrightarrow{\mathtt{rs}?\mathtt{a}} s$, with $\mathtt{s} \notin \{\mathtt{h}_i\}_{i\in I}$. Then there is no move in a gateway and also the channels between gateways are untouched. Hence, by definition of projection (Def. A.1), $s_{x_{|_\mathbb{K}}} = s_{|_\mathbb{K}}$. In particular, $q_{x_{\mathtt{h}_i}} = q_{\mathtt{h}_i} \notin \widehat{Q_{\mathtt{h}_i}}$ for all $i \in I$. Then, by the induction hypothesis for s_x, we have $s_{x_{|_\mathbb{K}}} \in \mathsf{RC}(\mathbb{K})$ and thus $s_{|_\mathbb{K}} \in \mathsf{RC}(\mathbb{K})$.

$s_x \xrightarrow{\mathtt{rs}!\mathtt{a}} s$, with $\mathtt{r} \notin \{\mathtt{h}_i\}_{i\in I}$.

This case is treated as the previous one.

$s_x \xrightarrow{\mathtt{rh}_i?\mathtt{a}} s$, with $i \in I$.

This case cannot apply, since, by definition of gateway, we would get $q_\mathtt{h} \in \widehat{Q}_\mathtt{h}$ which contradicts our assumption.

$s_x \xrightarrow{\mathtt{h}_i\mathtt{s}!\mathtt{a}} s$, with $i \in I$ and $\mathtt{s} \notin \{\mathtt{h}_i\}_{i\in I}$.

Then, by Lemma A.3(iii), there exist transitions $s_y \xrightarrow{\mathtt{rh}_i?\mathtt{a}} s_x \xrightarrow{\mathtt{h}_i\mathtt{s}!\mathtt{a}} s$ such that $s_y = (\vec{q}_y, \vec{w}_y) \in \mathsf{RC}(S)$ is reachable from s_0 in $n-2$ steps. Moreover, by definition of gateways and since $\mathtt{s} \notin \{\mathtt{h}_i\}_{i\in I}$, we have $\mathtt{r} = \mathtt{h}_j$ for some $j \in I \setminus \{i\}$. Hence, $s_y \xrightarrow{\mathtt{h}_j\mathtt{h}_i?\mathtt{a}} s_x \xrightarrow{\mathtt{h}_i\mathtt{s}!\mathtt{a}} s$ where

$$(q_{y_{\mathtt{h}_i}}, \mathtt{h}_j\mathtt{h}_i?\mathtt{a}, \hat{q}), (\hat{q}, \mathtt{h}_i\mathtt{s}!\mathtt{a}, q_{\mathtt{h}_i}) \in \delta_{\mathtt{h}_i} \quad \text{and} \quad w_{y_{\mathtt{h}_j\mathtt{h}_i}} = \mathtt{a} \cdot w_{\mathtt{h}_j\mathtt{h}_i}.$$

Then, by definition of gateway and connection policy,

$$(\dot{q}_{y_{\mathtt{h}_i}}, \mathtt{k}_j\mathtt{k}_i?\mathtt{a}, \dot{q}_{\mathtt{h}_i}) \in \delta^{\mathbb{K}}_{\mathtt{k}_i}.$$

Since, by definition of gateway, we have $q_{y_{\mathtt{h}_i}} \notin \widehat{Q_{\mathtt{h}_i}}$ and since $q_{y_{\mathtt{h}_k}} = q_{\mathtt{h}_k} \notin \widehat{Q_{\mathtt{h}_k}}$ for all $k \in I \setminus \{i\}$ (by assumption), we have also that $q_{y_{\mathtt{h}_k}} \notin \widehat{Q_{\mathtt{h}_k}}$ for each $k \in I$. Then we can apply the induction hypothesis for s_y and get $s_{y_{|_\mathbb{K}}} \in \mathsf{RC}(S)$.

If $s_{y_{|_\mathbb{K}}} = (\vec{q'}, \vec{w'})$ and $s_{|_\mathbb{K}} = (\vec{q''}, \vec{w''})$ then, by definition of projection and the above, we have

$$q'_{\mathtt{k}_i} = \dot{q}_{y_{\mathtt{h}_i}} \quad q''_{\mathtt{k}_i} = \dot{q}_{\mathtt{h}_i} \quad w'_{\mathtt{k}_j\mathtt{k}_i} = w_{y_{\mathtt{h}_j\mathtt{h}_i}} = \mathtt{a} \cdot w_{\mathtt{h}_j\mathtt{h}_i} = \mathtt{a} \cdot w''_{\mathtt{k}_j\mathtt{k}_i}.$$

Since $(\dot{q}_{y_{\mathtt{h}_i}}, \mathtt{k}_j\mathtt{k}_i?\mathtt{a}, \dot{q}_{\mathtt{h}_i}) \in \delta^{\mathbb{K}}_{\mathtt{k}_i}$, we get $s_{y_{|_\mathbb{K}}} \xrightarrow{\mathtt{k}_j\mathtt{k}_i?\mathtt{a}} s_{|_\mathbb{K}}$ and thus $s_{|_\mathbb{K}} \in \mathsf{RC}(\mathbb{K})$.

$s_y \xrightarrow{\mathtt{h}_i\mathtt{h}_j!\mathsf{a}} s$ with $i, j \in I, i \neq j$.

Then, by Lemma A.3(iii), there exist transitions $s_y \xrightarrow{\mathtt{rh}_i?\mathsf{a}} s_x \xrightarrow{\mathtt{h}_i\mathtt{h}_j!\mathsf{a}} s$ such that $s_y = (\vec{q}_y, \vec{w}_y) \in \mathsf{RC}(S)$ is reachable from s_0 in $n-2$ steps. Moreover, by definition of gateways and since $\mathtt{h}_j \in \{\mathtt{h}_i\}_{i\in I}$, we have $\mathtt{r} \notin \{\mathtt{h}_i\}_{i\in I}$ and

$$(q_{y_{\mathtt{h}_i}}, \mathtt{rh}_i?\mathsf{a}, \hat{q}), (\hat{q}, \mathtt{h}_i\mathtt{h}_j!\mathsf{a}, q_{\mathtt{h}_i}) \in \delta_{\mathtt{h}_i} \quad \text{and} \quad w_{\mathtt{h}_i\mathtt{h}_j} = w_{y_{\mathtt{h}_i\mathtt{h}_j}} \cdot \mathsf{a}$$

Then, by definition of gateway and connection policy,

$$(\dot{q}_{y_{\mathtt{h}_i}}, \mathtt{k}_i\mathtt{k}_j!\mathsf{a}, \dot{q}_{\mathtt{h}_i}) \in \delta^{\mathbb{K}}_{\mathtt{k}_i}.$$

Since, by definition of gateway, we have $q_{y_{\mathtt{h}_i}} \notin \widehat{Q_{\mathtt{h}_i}}$ and since $q_{y_{\mathtt{h}_k}} = q_{\mathtt{h}_k} \notin \widehat{Q_{\mathtt{h}_k}}$ for all $k \in I \setminus \{i\}$ (by assumption), we have also that $q_{y_{\mathtt{h}_k}} \notin \widehat{Q_{\mathtt{h}_k}}$ for each $k \in I$. Then we can apply the induction hypothesis for s_y and get $s_{y_{|\mathbb{K}}} \in \mathsf{RC}(S)$.

If $s_{y_{|\mathbb{K}}} = (\vec{q}', \vec{w}')$ and $s_{|\mathbb{K}} = (\vec{q}'', \vec{w}'')$ then, by definition of projection and the above, we have

$$q'_{\mathtt{k}_i} = \dot{q}_{y_{\mathtt{h}_i}} \quad q''_{\mathtt{k}_i} = \dot{q}_{\mathtt{h}_i} \quad w''_{\mathtt{k}_i\mathtt{k}_j} = w_{\mathtt{h}_i\mathtt{h}_j} = w_{y_{\mathtt{h}_i\mathtt{h}_j}} \cdot \mathsf{a} = w'_{\mathtt{k}_i\mathtt{k}_j} \cdot \mathsf{a}.$$

Since $(\dot{q}_{y_{\mathtt{h}_i}}, \mathtt{k}_i\mathtt{k}_j!\mathsf{a}, \dot{q}_{\mathtt{h}_i}) \in \delta^{\mathbb{K}}_{\mathtt{k}_i}$, we get $s_{y_{|\mathbb{K}}} \xrightarrow{\mathtt{k}_i\mathtt{k}_j!\mathsf{a}} s_{|\mathbb{K}}$ and thus $s_{|\mathbb{K}} \in \mathsf{RC}(\mathbb{K})$.

□

Lemma A.6. *Let $s = (\vec{q}, \vec{w}) \in \mathsf{RC}(S)$. Then there exists $s' = (\vec{q}', \vec{w}') \in \mathsf{RC}(S)$ such that $s \to^* s'$, $|w_{\mathtt{pq}}| \leq |w'_{\mathtt{pq}}|$ for all $\mathtt{pq} \in C$ and, for each $i \in I$, $q'_{\mathtt{h}_i} \notin \widehat{Q_{\mathtt{h}_i}}$ and $(q_{\mathtt{h}_i} \notin \widehat{Q_{\mathtt{h}_i}} \implies q'_{\mathtt{h}_i} = q_{\mathtt{h}_i})$.*

Proof. If $q_{\mathtt{h}_i} \in \widehat{Q_{\mathtt{h}_i}}$ for each $i \in I$, we are done by setting $s' = s$. Otherwise, given an $i \in I$ such that $q_{\mathtt{h}_i} \in \widehat{Q_{\mathtt{h}_i}}$, by Fact A.2(1) we can infer that there exists a configuration transition of the form

$$s \xrightarrow{\mathtt{h}_k\mathtt{s}!} s''$$

such that, for $s'' = (\vec{q}'', \vec{w}'')$, it holds that $|w_{\mathtt{pq}}| \leq |w''_{\mathtt{pq}}|$ for all $\mathtt{pq} \in C$, $q''_{\mathtt{h}_i} \notin \widehat{Q_{\mathtt{h}_i}}$ and, for each $j \in I$ with $j \neq i$, $q_{\mathtt{h}_j} \notin \widehat{Q_{\mathtt{h}_j}} \implies q''_{\mathtt{h}_j} = q_{\mathtt{h}_j}$. By iterating this sort of transitions, we get the thesis. □

In the following subsections we proceed to prove the preservation by multicomposition of the various communication properties separately.

A.2 Preservation of deadlock-freeness

Lemma A.7 (Projections of deadlocks are deadlocks)**.** *Let $S = \mathscr{MC}(\{S_i\}_{i\in I}, \mathbb{K})$ such that the interfaces of each S_i, i.e. the CFSMs M_{h_i} in S_i, have no mixed state. Let $s = (\vec{q}, \vec{w}) \in \mathsf{RC}(S)$ be a deadlock configuration of S. Then either*

- *there exists $i \in I$ such that $s_{|_\mathbf{i}} \in \mathsf{RC}(S_i)$ and $s_{|_\mathbf{i}}$ is a deadlock configuration of S_i; or*
- *$s_{|\mathbb{K}} \in \mathsf{RC}(\mathbb{K})$ and $s_{|\mathbb{K}}$ is a deadlock configuration of $\mathbb{K}$.*

Proof. By definition of deadlock configuration we have that, for each $i \in I$, $q_{\mathtt{h}_i} \notin \widehat{Q_{\mathtt{h}_i}}$. Otherwise there would exist a $j \in I$ such that $q_{\mathtt{h}_i} \in \widehat{Q_{\mathtt{h}_i}}$. But then, by Fact A.2(1), there would be an output transition from $q_{\mathtt{h}_j}$, contradicting s to be a deadlock configuration of S. So, by Proposition A.5, we get $s_{|_\mathbf{i}} \in \mathsf{RC}(S_i)$ for each $i \in I$, as well as $s_{|\mathbb{K}} \in \mathsf{RC}(\mathbb{K})$.

Now, since $s = (\vec{q}, \vec{w})$ is a deadlock configuration of S, we have $\vec{w} = \vec{\varepsilon}$ and $\forall \mathtt{p} \in \mathbf{P}.\ q_{\mathtt{p}}$ is a receiving state. Hence, by definitions of gateway and multicomposition (Defs 4.11 and 4.14), we need to take into account the following cases concerning the shapes of the transitions from the various $q_{\mathtt{h}_i}$ in their respective $\delta_{\mathtt{h}_i}$, i.e. the gateway transitions of $\mathtt{h}_i$.

$\underline{\diamond}$ *There exists $v \in I$ and a transition $(q_{\mathtt{h}_v}, \mathtt{sh}_v?\_, \_)$ with some $\mathtt{s} \in \mathbf{P}_v$.*
By the no-mixed-state assumption, the CFSM $M_{\mathtt{h}_v}$ in S_v has no mixed state. Therefore, by definition of gateway, all the transitions from $q_{\mathtt{h}_v}$ in $\delta_{\mathtt{h}_v}$ are of the form $(q_{\mathtt{h}_v}, \mathtt{sh}_v?\_, \_)$ with $\mathtt{s} \in \mathbf{P}_v$.
Then we can infer, again from the definition of gateway, that all transitions from $q_{\mathtt{h}_v}$ in $\delta^v_{\mathtt{h}_v}$, i.e. transitions in M_{h_v}, are of the form $(q_{\mathtt{h}_v}, \mathtt{sh}_v?\_, \_)$. Hence we obtain that $s_{|_\mathbf{v}}$ is a deadlock configuration of S_v.

$\underline{\diamond}$ *For each $v \in I$ all transitions from $q_{\mathtt{h}_v}$ in $\delta_{\mathtt{h}_v}$ are of the form $(q_{\mathtt{h}_v}, \mathtt{h}'\mathtt{h}_v?\_, \_)$ with some $\mathtt{h}' \in \{\mathtt{h}_i\}_{i \in I \setminus \{v\}}$.*
W.l.o.g. we consider a generic single transition $(q_{\mathtt{h}}, \mathtt{h}'\mathtt{h}_v?a, \widehat{q})$. Now, by definition of gateway, we have that

$$(q_{\mathtt{h}_v}, \mathtt{h}'\mathtt{h}_v?a, \widehat{q}), (\widehat{q}, \mathtt{h}_v\mathtt{s}!a, q'_{\mathtt{h}_v}) \in \delta_{\mathtt{h}_v} \text{ for some } \mathtt{s} \in \mathbf{P}_v.$$

This implies that

$$(q_{\mathtt{h}_v}, \mathtt{h}_v\mathtt{s}!a, q'_{\mathtt{h}_v}) \in \delta^v_{\mathtt{h}_v}.$$

So, by definition of connection policy, we can infer that

$$(\dot{q}_{\mathtt{h}_v}, \mathtt{k}'\mathtt{k}_v?a, \dot{q}'_{\mathtt{h}_v}) \in \delta^{\mathbb{K}}_{\mathtt{k}_v}.$$

By definition of projection, $s_{|_\mathbb{K}} = (\vec{p}, \vec{w'})$ such that $p_{\mathtt{k}_v} = \dot{q}_{\mathtt{h}}$ and $w'_{\mathtt{k}_j\mathtt{k}_i} = w_{\mathtt{h}_j\mathtt{h}_i}$ for each $j, i \in I$ with $j \neq i$. Since $\vec{w} = \vec{\varepsilon}$ we obtain $\vec{w'} = \vec{\varepsilon}$ and $p_{\mathtt{k}_v}$ is a receiving state since $p_{\mathtt{k}_v} = \dot{q}_{\mathtt{h}}$. Hence, $s_{|_\mathbb{K}} \in \mathsf{RC}(\mathbb{K})$ is a deadlock configuration of $\mathbb{K}$.

□

Corollary A.8 (Preservation of deadlock-freeness)**.** *Let $S = \mathscr{MC}(\{S_i\}_{i \in I}, \mathbb{K})$ such that, for each $i \in I$, S_i is deadlock-free and $\mathbb{K}$ is deadlock-free. Moreover, let the interfaces of each S_i, i.e. the CFSMs M_{h_i} in S_i, have no mixed state. Then S is deadlock-free.*

Proof. By contradiction, let us assume there is an $s \in \mathsf{RC}(S)$ which is a deadlock configuration of S. Then we get a contradiction by Lemma A.7. □

A.3 No-orphan-message preservation

Lemma A.9. *Let $s = (\vec{q}, \vec{w}) \in \mathsf{RC}(S)$ be an orphan-message configuration of S. Then either*

- *there exists $i \in I$ such that $s_{|_\mathbf{i}} \in \mathsf{RC}(S)$ and $s_{|_\mathbf{i}}$ is an orphan-message configuration of S_i; or*
- *$s_{|_\mathbb{K}} \in \mathsf{RC}(\mathbb{K})$ and $s_{|_\mathbb{K}}$ is an orphan-message configuration of $\mathbb{K}$.*

Proof. Let $s = (\vec{q}, \vec{w}) \in \mathsf{RC}(S)$ be an orphan-message configuration of S, that is $\vec{q}$ is final and $\vec{w} \neq \vec{\varepsilon}$. Since $\vec{q}$ is final we get, by Fact A.2(1, that, for each $i \in I$, $q_{\mathtt{h}_i} \notin \widehat{Q_{\mathtt{h}_i}}$. So, by Proposition A.5, $s_{|_\mathbb{K}} \in \mathsf{RC}(\mathbb{K})$ and $s_{|_\mathbf{i}} \in \mathsf{RC}(S_i)$ for each $i \in I$. We have to consider now only the following two cases:

$\exists i \in I. \exists \mathtt{p}, \mathtt{q} \in \mathbf{P}_i$ such that $w_{\mathtt{pq}} \neq \varepsilon$.
In such a case we immediately get, by definition of projection, that $s_{|_\mathbf{i}}$ is an orphan-message configuration of S_i.

$\exists \mathtt{p},\mathtt{q} \in \{\mathtt{h}_i\}_{i\in I}$ with $w_{\mathtt{pq}} \neq \varepsilon$

Let $s_{|\mathbb{K}} = (\vec{q'},\vec{w'})$. Since $\vec{q}$ is final, by definition of projection, $\vec{q'}$ is final as well. Moreover, we have that $w'_{\mathtt{k}_j\mathtt{k}_v} = w_{\mathtt{h}_j\mathtt{h}_v}$, for each $j,v \in I$ with $j \neq v$. This implies that $\vec{w'} \neq \vec{\varepsilon}$. Hence, $s_{|\mathbb{K}}$ is an orphan-message configuration of $\mathbb{K}$.

□

Corollary A.10 (Preservation of orphan-message-freeness)**.** *Let $S = \mathcal{MC}(\{S_i\}_{i\in I},\mathbb{K})$ such that, for each $i \in I$, S_i is orphan-message free and also $\mathbb{K}$ is orphan-message free. Then is S orphan-message free.*

Proof. By contradiction, let us assume there is an $s \in \mathsf{RC}(S)$ which is an orphan-message configuration. Then we get a contradiction by Lemma A.9. □

A.4 Preservation of no unspecified reception

Proposition A.11 (Preservation of reception-error-freeness)**.** *Let $S = \mathcal{MC}(\{S_i\}_{i\in I},\mathbb{K})$ such that, for each $i \in I$, no element in $\mathsf{RC}(S_i)$ and also no element in $\mathsf{RC}(\mathbb{K})$ is an unspecified reception configuration. Moreover, let the interfaces of each S_i, i.e. the CFSMs M_{h_i} in S_i, have no mixed state. Then there is no unspecified reception configuration in $\mathsf{RC}(S)$.*

Proof. By contradiction, let us assume there is an $s = (\vec{q},\vec{w}) \in \mathsf{RC}(S)$ which is an unspecified reception configuration. So, let $\mathtt{r} \in \mathbf{P}$ and let $q_\mathtt{r}$ be the receiving state of $M_\mathtt{r}$ prevented from receiving any message from any of its buffers, which all are not empty (Definition 3.4(iii)). In particular, let $z \in I$ such that $\mathtt{r} \in \mathbf{P}_z$. Now we take into account the following possible cases where, for the sake of readability, we set $\mathtt{h} = \mathtt{h}_z$:

$q_\mathtt{h} \notin \widehat{Q_\mathtt{h}}$.

By Proposition A.5(i) we get $s_{|\mathtt{z}} \in \mathsf{RC}(S_z)$. We distinguish now two further subcases.

$\mathtt{r} \neq \mathtt{h}$

Then $s_{|\mathtt{z}} \in \mathsf{RC}(S_z)$ is an unspecified reception configuration of S_z. Contradiction!

$\mathtt{r} = \mathtt{h}$

Since $q_\mathtt{r} (= q_\mathtt{h})$ is a receiving state, by definition of gateway it follows that the set of all outgoing transitions from $q_\mathtt{h}$ in $\delta_\mathtt{h}$ is of the form

$$\{(q_\mathtt{h}, \mathtt{s}_j\mathtt{h}?\mathtt{a}_j, \widehat{q_j})\}_{j=1..m}$$

where $\widehat{q_j} \in \widehat{Q_\mathtt{h}}$ and $\mathtt{s}_j \in \mathbf{P}_z \cup \{\mathtt{h}_i\}_{i\in I}, \mathtt{s}_j \neq \mathtt{h}$.

By definition of unspecified reception configuration, we have hence that for all $j = 1..m$,

$$\mid w_{\mathtt{s}_j\mathtt{h}} \mid > 0 \text{ and } w_{\mathtt{s}_j\mathtt{h}} \notin \mathtt{a}_j \cdot \mathbb{A}^* \tag{1}$$

Now, due to the no-mixed-state assumption, it suffices to consider the following two possibilities:

⋄ *For each $j = 1..m$, $\mathtt{s}_j \notin \{\mathtt{h}_i\}_{i\in I}$.*

In this case we can infer from Fact A.23b) that, for each $j = 1..m$, there exists $q'_j \in Q_\mathtt{h}$ such that $(q_\mathtt{h}, \mathtt{s}_j\mathtt{h}?\mathtt{a}_j, q'_j) \in \delta^z_\mathtt{h}$. This implies that $s_{|\mathtt{z}} \in \mathsf{RC}(S_z)$ is an unspecified reception configuration of S_z. Contradiction!

$\underline{\diamond}$ *For each $j = 1..m$, there exists $v_j \in I$ such that $\mathtt{s}_j = \mathtt{h}_{v_j}$.*

In this case, by definition of gateway, we can infer that, for each $j = 1..m$,

$$(q_{\mathtt{h}}, \mathtt{h}_{v_j}\mathtt{h}?\mathtt{a}_j, \widehat{q}_j), (\widehat{q}_j, \mathtt{h}\mathtt{u}_j!\mathtt{a}_j, q'_j) \in \delta_{\mathtt{h}}$$

where $\mathtt{u}_j \in \mathbf{P}_z, q'_j \in Q_{\mathtt{h}}$ and $(q_{\mathtt{h}}, \mathtt{h}\mathtt{u}_j!\mathtt{a}_j, q'_j) \in \delta^{z}_{\mathtt{h}}$.

Moreover, by definition of multi-composition w.r.t. connection policy $\mathbb{K}$, we have that, for each $j = 1..m$,

$$(\dot{q}_{\mathtt{h}}, \mathtt{k}_{v_j}\mathtt{k}_z?\mathtt{a}_j, \dot{q}'_j) \in \delta^{\mathbb{K}}_{\mathtt{k}_z}$$

We consider now the possible two subcases.

$-$ $q_{\mathtt{h}_i} \notin \widehat{Q_{\mathtt{h}_i}}$ for each $i \in I$.

Then, by Proposition A.5(ii) we get $s_{|\mathbb{K}} \in \mathsf{RC}(\mathbb{K})$. If $s_{|\mathbb{K}} = (\vec{p}, \vec{w'})$, by definition of projection we have that $p_{\mathtt{k}_z} = \dot{q}_{\mathtt{h}}$ and $w'_{\mathtt{k}_x\mathtt{k}_y} = w_{\mathtt{h}_x\mathtt{h}_y}$ for each $x, y \in I$ such that $x \neq y$. The last equalities and (1) imply that, for each $j = 1..m$,

$$\mid w_{\mathtt{k}_{v_j}\mathtt{k}_z} \mid > 0 \text{ and } w_{\mathtt{k}_{v_j}\mathtt{k}_z} \notin \mathtt{a}_j \cdot \mathbb{A}^*.$$

Thus $s_{|\mathbb{K}} \in \mathsf{RC}(\mathbb{K})$ is an unspecified reception configuration of $\mathbb{K}$. Contradiction!

$-$ There exists $k \in I$ such that $q_{\mathtt{h}_k} \in \widehat{Q_{\mathtt{h}_k}}$.

By Lemma A.6, there exists $s' = (\vec{q''}, \vec{w''}) \in \mathsf{RC}(S)$ such that $s \to^* s'$, $|w_{\mathtt{pq}}| \leq |w''_{\mathtt{pq}}|$ for all $\mathtt{pq} \in C$, $q''_{\mathtt{h}_j} \notin \widehat{Q_{\mathtt{h}_j}}$ for each $j \in I$, and $q''_{\mathtt{h}} = q_{\mathtt{h}}$ since we have assumed $q_{\mathtt{h}} \notin \widehat{Q_{\mathtt{h}}}$. By Proposition A.5(ii) we get $s'_{|\mathbb{K}} \in \mathsf{RC}(\mathbb{K})$. By reasoning as in the previous case, we get that $s'_{|\mathbb{K}}$ is an unspecified reception configuration of $\mathbb{K}$. Contradiction!

$q_{\mathtt{h}} \in \widehat{Q_{\mathtt{h}}}$.

Then, by Fact A.2(1), $q_{\mathtt{h}} \in \widehat{Q_{\mathtt{h}}}$ is a sending state such that $(q_{\mathtt{h}}, \mathtt{hs}!\mathtt{a}, q'_{\mathtt{h}}) \in \delta_{\mathtt{h}}$ with $q'_{\mathtt{h}} \in Q_{\mathtt{h}}$ and $\mathtt{s} \in \mathbf{P}_z \cup \{\mathtt{h}_i\}_{i \in I}$. Since $q_{\mathtt{r}}$ is a receiving state, it is impossible that $\mathtt{r} = \mathtt{h}$. So, let $\mathtt{r} \neq \mathtt{h}$. It is now immediate to check that there exists $s' \in \mathsf{RC}(S)$ such that $s \xrightarrow{\mathtt{hs}!a} s' = (\vec{q'}, \vec{w'})$ with $q'_{\mathtt{h}}$ and $\mathtt{s}$ as above. Since $q'_{\mathtt{h}} \notin \widehat{Q_{\mathtt{h}}}$ it follows, by Proposition A.5(i), that $s'_{|_{\mathbf{z}}} \in \mathsf{RC}(S_z)$. Moreover, we have that

a) $\forall \mathtt{p} \neq \mathtt{h}.\ q'_{\mathtt{p}} = q_{\mathtt{p}}$ and, in particular, $q'_{\mathtt{r}} = q_{\mathtt{r}}$;

b) $\forall \mathtt{pq} \neq \mathtt{hs}.\ w'_{\mathtt{pq}} = w_{\mathtt{pq}}$;

c) $w'_{\mathtt{hs}} = w_{\mathtt{hs}} \cdot \mathtt{a}$.

We consider now the following two possible subcases:

$\mathtt{s} = \mathtt{h}_v$ for some $v \in I$ with $v \neq z$.

Then $\mathtt{s} \neq \mathtt{r}$, since $\mathtt{r} \in \mathbf{P}_z$. From (a) and (b) above it follows that $q'_{\mathtt{r}} = q_{\mathtt{r}}$ and, since $\mathtt{s} \neq \mathtt{r}$, $w'_{\mathtt{pr}} = w_{\mathtt{pr}}$ for all $\mathtt{p} \in \mathbf{P}_z$. Consequently, $s'_{|_{\mathbf{z}}} \in \mathsf{RC}(S_z)$ is an unspecified reception configuration of S_z. Contradiction!

$\mathtt{s} \in \mathbf{P}_z$

If $\mathtt{s} \neq \mathtt{r}$ we get a contradiction as in the previous case.

If $\mathtt{s} = \mathtt{r}$, then $\mathtt{h}$ sends the message $\mathtt{a}$ to the buffer $w_{\mathtt{hr}}$. Since $q_{\mathtt{r}}$ is the receiving state of $M_{\mathtt{r}}$ prevented from receiving any message from any of its buffers, which all are not empty in configuration s, the sending of $\mathtt{a}$ extends $w_{\mathtt{hr}}$ which still has a wrong element on its first position. Then, by (a) and (b) above $s'_{|_{\mathbf{z}}}$ is an unspecified reception configuration of S_z. Contradiction!

□

A.5 Progress preservation

Proposition A.12 (Progress preservation)**.** *Let $S = \mathscr{MC}(\{S_i\}_{i\in I}, \mathbb{K})$ such that $\mathbb{K}$ and each S_i (for all $i \in I$) do enjoy the progress property. Moreover, let the interfaces of each S_i, i.e. the CFSMs M_{h_i} in S_i, have no mixed state. Then also S does enjoy the progress property.*

Proof. The proof is performed by contradiction. Let us assume S not to enjoy the progress property, namely that there exists $s = (\vec{q}, \vec{w}) \in \mathsf{RC}(S)$ such that

$$s \not\longrightarrow \quad \text{and} \quad \vec{q} \text{ is not final, i.e. } \exists \mathtt{r} \in \mathbf{P}.\ q_{\mathtt{r}} \text{ is not final in } M_{\mathtt{r}}. \tag{2}$$

In particular, there exists $z \in I$ and $\mathtt{r} \in \mathbf{P}_z$ such that $q_{\mathtt{r}}$ is not final in $M_{\mathtt{r}}$.

By $s \not\longrightarrow$ and by Fact A.2(1), we have that for each $i \in I$, $q_{\mathtt{h}_i} \notin \widehat{Q_{\mathtt{h}_i}}$. Otherwise there would be an output transition from some $q_{\mathtt{h}_i}$, contradicting $s \not\longrightarrow$. So, by Proposition A.5, we get $s_{|_\mathbf{i}} \in \mathsf{RC}(S_i)$ for each $i \in I$, as well as $s_{|_\mathbb{K}} \in \mathsf{RC}(\mathbb{K})$. In particular, $s_{|_\mathbf{z}} \in \mathsf{RC}(S_z)$.

Now we distinguish two cases:

Case 1: $q_{\mathtt{h}_z}$ is final in $M_{\mathtt{h}_z}$.

Then $q_{\mathtt{h}_z}$ is final in $M^z_{\mathtt{h}_z}$. Consequently, $s \not\longrightarrow$ and $s_{|_\mathbf{z}} \in \mathsf{RC}(S_z)$ implies $s_{|_\mathbf{z}} \not\longrightarrow$. Moreover, we know that $q_{\mathtt{r}}$ is not final in $M_{\mathtt{r}}$ which is an element of the system S_z. Hence, S_z does not enjoy the progress property. Contradiction!

Case 2: $q_{\mathtt{h}_z}$ is not final in $M_{\mathtt{h}_z}$.

From above we know $q_{\mathtt{h}_z} \notin \widehat{Q_{\mathtt{h}_z}}$ and hence, by construction of gateways, there must be an input transition in $M_{\mathtt{h}_z}$ starting in $q_{\mathtt{h}_z}$. We distinguish two subcases:

Case 2.1: There exists a transition from $q_{\mathtt{h}_z}$ in $\delta_{\mathtt{h}_z}$ of the form $(q_{\mathtt{h}_z}, \mathtt{s}\mathtt{h}_z?\_, \_)$ with $\mathtt{s} \in \mathbf{P}_z$.

By the no-mixed-state assumption and by construction of gateways it follows that all transitions in $\delta_{\mathtt{h}_z}$ with source state $q_{\mathtt{h}_z}$ have this form. Then (i) also all transitions in $\delta^z_{\mathtt{h}_z}$ with source state $q_{\mathtt{h}_z}$ have this form and (ii) there exists at least one such transition. Since $s \not\longrightarrow$ and $s_{|_\mathbf{z}} \in \mathsf{RC}(S_z)$ we obtain, as a consequence of (i), that $s_{|_\mathbf{z}} \not\longrightarrow$. As a consequence of (ii) $q_{\mathtt{h}_z}$ is also not final in $M^z_{\mathtt{h}_z}$. Thus S_z does not enjoy the progress property. Contradiction!

Case 2.2: There exists a transition from $q_{\mathtt{h}_z}$ in $\delta_{\mathtt{h}_z}$ of the form $(q_{\mathtt{h}_z}, \mathtt{h}_j\mathtt{h}_z?\_, \_)$ with $j \in I$.

By the no-mixed-state assumption and by construction of gateways it follows that all transitions in $\delta_{\mathtt{h}_z}$ with source state $q_{\mathtt{h}_z}$ have this form. Then, by definition of gateway and connection policy, all transitions in $\delta^{\mathbb{K}}_{\mathtt{k}_z}$ have the form $(\dot{q}_{\mathtt{k}_z}, \mathtt{k}_j\mathtt{k}_z?\_, \_)$ and there exists at least one such transition. Let now $\vec{s}_{|_\mathbb{K}} = (\vec{q'}, \vec{w'})$. By definition of projection, $w'_{\mathtt{k}_j\mathtt{k}_z} = w_{\mathtt{h}_j\mathtt{h}_z}$ for each $j, z \in I, j \neq z$. So, since $s \not\longrightarrow$ and $s_{|_\mathbb{K}} \in \mathsf{RC}(\mathbb{K})$, we can infer that $\vec{s}_{|_\mathbb{K}} \not\longrightarrow$. Moreover, $\dot{q}_{\mathtt{k}_z}$ is not final in $M_{\mathtt{k}_z}$. Thus $S_\mathbb{K}$ does not enjoy the progress property. Contradiction!

□