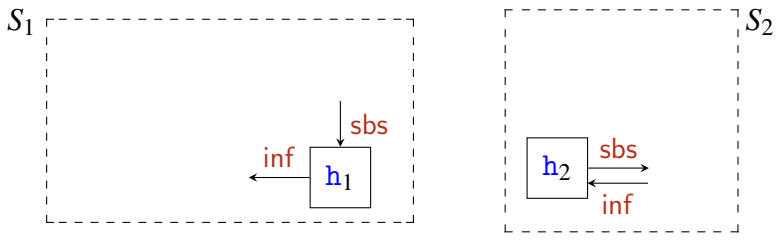
# Safe Composition of CFSM Systems via Partial Gateways

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

The *Participants-as-Interfaces* (PaI) methodology for system composition proposes that system participants can be regarded as interfaces. For each system in a given collection, one participant is designated to serve as its interface. When the systems are composed, these interface participants are replaced with gateways that communicate with one another by forwarding messages. We generalise the approach to *partial gateways*, where gateways can forward only a choosen set of messages. As for the standard PaI approach, we exploit such extended version for systems of communicating finite state machines (CFSMs). This extension is fully detailed for the binary case, and can be scaled up to the multicomposition case. We prove that many relevant communication properties (deadlock-freeness, reception-error-freeness, etc.) are preserved by *PaI composition via partial gateways* in case also the connection policy (i.e. the system representing the way we wish gateways interact with each other) enjoys the same properties. Such a proof turns out to be just a corollary of a property preservation result for a restricted and partial composition method, dubbed *fusion-composition*. Fusion-composition hence turns out to be at the heart of the PaI composition approach.

## 1 Introduction

PaI is an approach to the composition of concurrent/distributed systems introduced in [3, 4] and further investigated in papers among which [8, 9, 7, 6, 2, 10]. It is specifically designed for systems that communicate via message passing, in which the communication behaviours of participants can also be interpreted as interfaces. Our intended meaning of "interface" (a term actually used in the literature with several different connotations) is, informally, a description of the behaviour of an outer system. In the drawing (1) below we sketch two systems $S_1$ and $S_2$. For the sake of simplicity and to focus only on the most relevant issues, we now abstract the participants' behaviours from everything but the possibility of sending/receiving messages in such a drawing and throughout the present introduction. In particular, we abstract away from dynamic issues such as the logical order of the exchanged messages, the representation of which depends on the chosen formalism. System $S_1$ has a participant, $\mathtt{h}_1$, which can receive and send messages, sbs and inf, respectively, from and to other participants of $S_1$, say $\mathtt{r}$ and $\mathtt{r}'$. Meanwhile, in $S_2$, $\mathtt{h}_2$ can send and receive messages, sbs and inf, respectively, to and from another participant of $S_2$, say $\mathtt{s}$. For simplicity, only participants $\mathtt{h}_1$ and $\mathtt{h}_2$ are shown inside the dashed boxes representing systems $S_1$ and $S_2$, respectively.




(1)

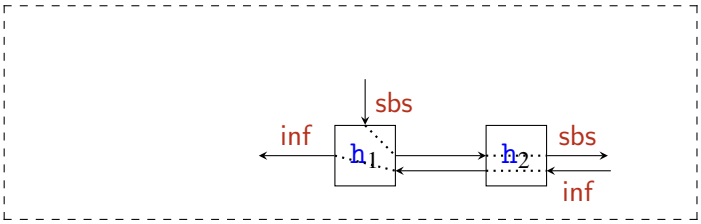
*Partially supported by Project "National Center for HPC, Big Data e Quantum Computing", Programma M4C2, Investimento 1.3.; and by Project PIA.CE.RI (PIAno di inCEntivi per la RIcerca di Ateneo) UniCT 2024/2026.

To appear in EPTCS.

$S_1$ may represent a domotic system in which component $\mathtt{h}_1$ controls the diffusion of a given substance upon receiving the "set-by-sensor" message sbs from either $\mathtt{r}$ or $\mathtt{r}'$ (the sensors). Through the message inf, $\mathtt{h}_1$ can also provide the other participants with some informations about the effects of the diffusion. $S_2$, instead, can be seen as an aroma diffusion system where, following a request from a sensor $\mathtt{h}_2$ participant $\mathtt{s}$ coordinates the release of an aromatic substance. Participant $\mathtt{h}_2$ may likewise receive information about the resulting effects.

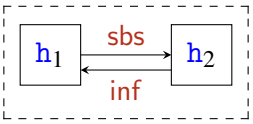
Now, rather than interpreting $\mathtt{h}_1$ as the behaviour of a participant of $S_1$, we could interpret it as the behaviour of an outer system interacting with $S_1$ by sending the message sbs and receiving inf. In other words, it can be viewed as an *interface*, in the previosly hinted at sense. Analogously, $\mathtt{h}_2$ can be viewed as an external system interacting with $S_2$ by receiving sbs messages and sending inf. The idea underlying *PaI composition* is that, once two participants are identified as interfaces according to our current needs, they are replaced by forwarders, referred to as "gateways". In our simple example, the resulting system would look like (2) below, where dotted lines indicate message forwarding. The gateway that now replaces $\mathtt{h}_1$ (while retaining the same name) forwards any sbs messages received from the other participants of $S_1$, to the gateway that has replaced $\mathtt{h}_2$ (which also retains the same name). Upon receiving such a message, the gateway $\mathtt{h}_2$ forwards it to $\mathtt{s}$. The process is symmetrical for the inf messages received by $\mathtt{h}_2$.

$S_1 \overset{\mathtt{h}_1 \leftrightarrow \mathtt{h}_2}{} S_2$




(2)

Note that any participant can be treated as an *interface*. This makes the approach suitable for the composition of closed systems. However, some conditions are required for the chosen interfaces in order to ensure that PaI composition is safe, i.e. property preserving. These conditions clearly depend on the formalism used to describe participant behaviours. Several results have been established using the automata-based formalism of Communicating Finite State Machines (CFSM) [11]. It has been shown in [8, 9] that a given communication property $P$ (e.g. deadlock freedom and orphan-message freedom among others) is preserved by composition if the connection policy used for the composition also enjoys $P$. A *connection policy* is intended as a description of of how messages should be forwarded by the gateways in a composition and it can be formalised in terms of another system. In our present simple example a natural connection policy would consist of a two-CFSM system, which is abstractly represented as follows.

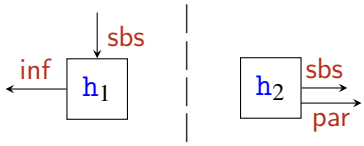



(3)

In PaI binary composition, there is always just one connection policy since the participant to which a message can be forwarded is uniquely determined. Therefore, the gateways that we substitute for interfaces are also uniquely determined. The results in [8, 9] consider the composition of more than two systems, where the forwarding policies among several interface participants can be chosen with greater freedom. It is worth noting that the notion of a connection policy and the aforementioned property preservation results do not depend on any "duality" relationship between the chosen interfaces for the composition. This means that we could choose interfaces and connection policies that result in gateways,

some of which cannot actually exchange messages, although they are forwarded or expected by one of the gateways. Although this unseemly conduct does not necessarily hinder property preservation results, it can be prevented by imposing connection policies that comply with *connection models*, as defined in references [8, 9]. (We do not address this notion in the present paper). Alternatively, one could argue that certain messages, although present in an interface, should not be forwarded by the gateways at all.

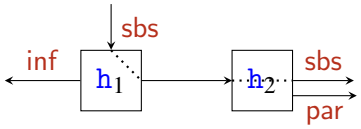
In the present paper, partly inspired by [5], we generalise the PaI approach by considering participants as interfaces only in part. Let us consider the following example. (From now on, unless necessary, we will avoid representing systems by dashed boxes and will only represent the participants identified as interfaces. Their belonging to different systems is indicated by dashed vertical lines.)

$S_1$ 

$S_2$ (4)

System $S_1$ remains as before, whereas in system $S_2$ the participant $\mathtt{h}_2$ can, in addition to sending the message sbs, also provide parameters for properly adjusting the diffusion of the aromatic substance via the message par. In this example, substituting $\mathtt{h}_1$ and $\mathtt{h}_2$ with uniformly built gateways, as required by the PaI approach, would result rather unreasonable. The gateway for $\mathtt{h}_1$ would never be able to emit inf since, as a forwarder, it would never be able to receive it from the gateway for $\mathtt{h}_2$. This might disrupt most of the communication properties that the two systems satisfy.

A more reasonable approach would be to consider only the actions concerning the message sbs as "interface actions", leaving the interpretation of the others as pertaining to their respective participants. Following this approach, we would replace the interface participants with *partial gateways* that only forward messages relating to interface actions, as informally shown below.



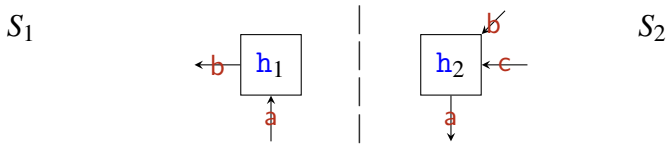
(5)

In this paper, we extend the PaI approach to include the composition of systems via partial gateways and identify the conditions under which this is applicable. We achieve this in the context of the CFSM formalism. The safety of the approach is proven by resorting to a restricted and simplified form of composition that we refer to as *partial-fusion composition*.

*Overview.* In Section 2 we present an informal description of PaI composition via partial gateways and show how this can be derived from the restricted form of composition we refer to as partial-fusion composition. The CFSM formalism is described in Section 3, where some relevant communication properties are also defined. A complete formalisation of the PaI composition via partial gateways in the CFSM setting is provided in Section 4. Before their formalisation, the underlying notions will be introduced and discussed by means of a running example. The property preservation result for PaI composition via partial gateways is stated at the end of that section. Section 5 is devoted to the formalisation of partial-fusion composition and how to obtain PaI composition from it. The latter's property preservation result can therefore be obtained by applying a similar result for the former. A concluding section briefly summarises the paper and provides some guidelines for future work.

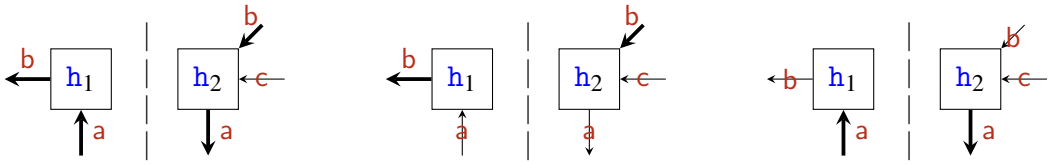
## 2 Informal description of PaI composition with partial gateways.

While maintaining an abstract level of description, we now elaborate on the idea of partial gateways mentioned in the introduction, using a further example. By considering the possibility of forwarding only some messages in a composition, partial gateways are generally not uniquely determined by the interface participants chosen for a composition, even in the binary case. Consider, for example, the following interface participants for systems $S_1$ and $S_2$.



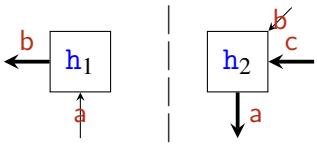

(6)

Message c cannot reasonably be considered one to be forwarded (although the usual binary PaI composition would make it). We have, however, definitely a choice concerning messages a and b. According to our intentions, we can decide whether they have to be exchanged between the partial gateways (i.e. whether they belong to the interface parts of $h_1$ and $h_2$) or not (i.e. whether the partial gateways will deal with them as $h_1$ and $h_2$ did prior to the composition). This type of decision can be represented graphically by using thicker lines for the actions intended for the interface parts of $h_1$ and $h_2$. The following reasonable choices are possible.




(7)

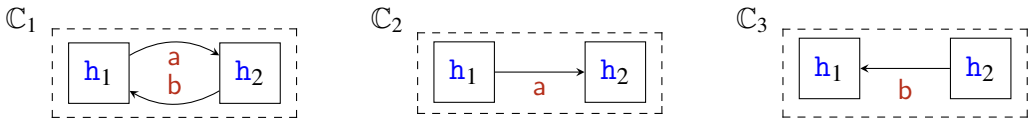
It is also possible to make choices that would hardly result in safe compositions.




(8)

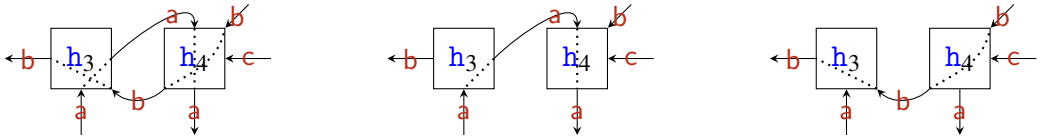
The notion of "participant with interface actions", hinted at above, will be formalised in terms of CFSMs, as *CFSM with interface transitions* (see Definition 4.1).

Our goal is to generalise the results in [8, 9], where the preservation of a property by composition is implied by the property also being satisfied by the connection policy used. In our setting, a connection policy determines how messages labelling interface actions are exchanged among partial gateways in the intended composition. The interface actions represented in (7) correspond to the three connection policies abstractly described as follows.



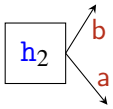

(9)

From the connection policy represented in (9) we get different compositions by means of the following partial gateways.

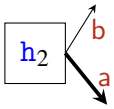



(10)

In general, not all choices of interface actions make sense. In some cases, it is clearly unreasonable to select an action as an interface action. For example, consider the message c in (6). However, there are other cases where the selection of an interface action is not sound for subtler reasons. Let us consider the participant $h_2$ in (11) below. Assume also that the CFSM specifying the dynamics of the behaviour of $h_2$ contains an actual branching between sending either a or b. In this case, the choice of interface actions described in (12) would not make sense.




(11)



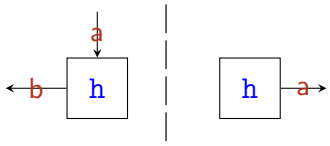

(12)

In fact, selecting the transmission of message a as interface action implies that when the interface participant $h_2$ is substituted with a partial gateway in a composition, the latter should decide whether to wait for a message a to be forwarded or to autonomously send the message b. Therefore, it is possible that the partial gateway sends message b, even though message a is about to arrive. Since CFSM is a formalism involving asynchronous communication, the message a might remain in the communication channel indefinitely once received, thereby disrupting some communication property of the composition. In particular, orphan-message freedom. It could be argued that the aforementioned problem actually involves the indirect presence of *mixed states* (states containing both input and output outgoing transitions). The presence of mixed states is often problematic in CFSM systems. However, we shall discuss other cases of problematic interface action selection that do not involve mixed states, neither directly in the interfaces nor indirectly in the partial gateways. In order to prevent issues like the one described above from occurring, we shall provide a syntactic condition that must be satisfied by the selection of interface actions.

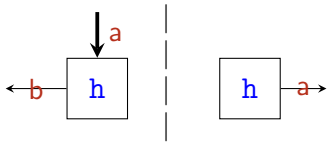
The above sketched PaI approach to binary composition via partial gateways naturally scales up to PaI multicomposition and PaI orchestrated multicomposition [8, 9]. However, for the sake of clarity and due to space limitations, we consider only the binary case.

One of the paper's main findings is that PaI composition via partial gateways preserves a number of communication properties if the connection policy used for the composition satisfies the same properties and some specific requirements due to partiality. For all the properties, but orphan-message-freedom, the no-mixed-state condition is required for interfaces. This result is obtained as a corollary of a preservation result for a restricted and simple form of binary PaI composition, which we call *partial-fusion composition*. All forms of PaI composition for asynchronous CFSMs that have been investigated so far, as well as the one via partial gateways that we introduce in this paper, can be obtained via a number of binary partial-fusion compositions. This composition can therefore be considered at the very core of the PaI approach to system composition.

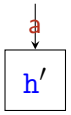
**Partial-Fusion Composition** This particular form of composition is binary. To get an intuition, let us consider the following simple example, in which both $S_1$ and $S_2$ have a participant named h that we select as their respective participant for the composition.

$S_1$  $S_2$ (13)

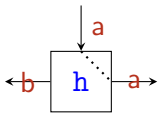

The composition method requires that some actions are choosen as interface actions in only one of the two participants so that, if possible, we get two participants that complement each other perfectly. For instance, by choosing

$S_1$  $S_2$ (14)


and "restricting" the behaviour of h in $S_1$ (the participant with interface actions) to its interface actions only, we obtain

 (15)

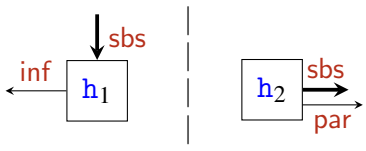

Now h′ above and h of $S_2$ in (14) are perfectly complementary (roughly, one has an input where the other has an output, and vice versa). Such a complementarity relation enables the h in $S_1$ and the h in $S_2$ represented in (13) to be partially *fused*. The fusion makes the two hs a single, partially forwarding gateway. Messages pertaining to interface actions are forwarded, while the partial gateway continues to behave as the h of $S_1$ with regard to non-interface actions.

$S_1 \overset{\text{h}}{\texttt{fuse}} S_2$  (16)

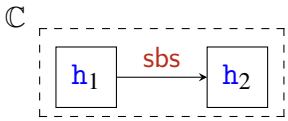

Notably, the relationship between fusible participants corresponds to that between interface participants and their respective participants in a connection policy within the PaI composition via partial gateways.

We now informally show how PaI composition via partial gateways can be achieved through a number of partial-fusion compositions. This is the primary objective of partial-fusion. From this perspective, the requirement to consider participants with the same name in the two systems is not as restrictive as it seems at first glance.

Consider the interface participants for $S_1$ and $S_2$ in (4) and the following selection of interface actions.

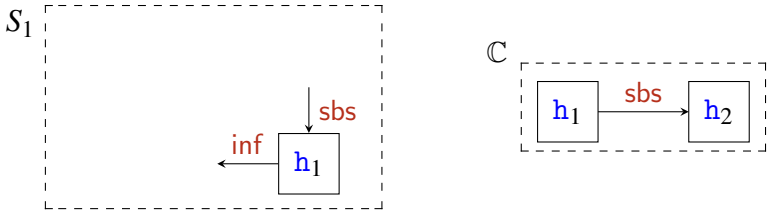
$S_1$  $S_2$ (17)


Since we are considering a binary case, the connection policy is uniquely determined by the selected interface actions.



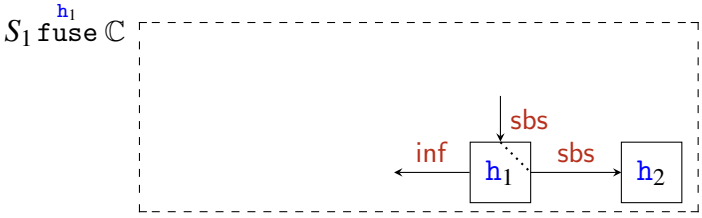

(18)

Now we consider systems $S_1$ and $\mathbb{C}$ (recall that here $\mathbb{C}$ represents a system of the same sort as $S_1$ and $S_2$).



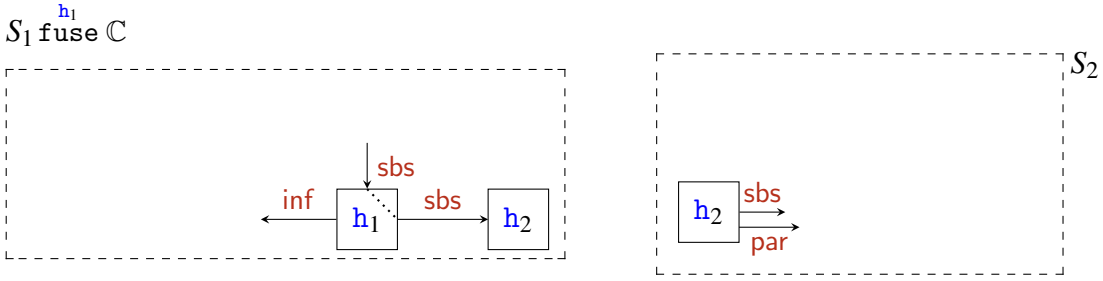

(19)

The above system are elegible for fusion composition, which returns the following system.



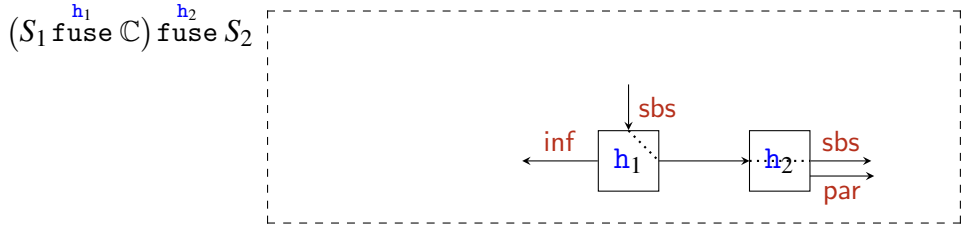

(20)

We now consider the following two systems.




(21)

By applying fusion composition again, we get the following system.




(22)

We have that $(S_1 \overset{h_1}{\texttt{fuse}} \mathbb{C}) \overset{h_2}{\texttt{fuse}} S_2 \equiv S_1 \overset{h_1 \leftrightarrow h_2}{} S_2$, where $S_1 \overset{h_1 \leftrightarrow h_2}{} S_2$ is the system obtained by PaI composition via partial gateways using the connection policy $\mathbb{C}$ as depicted in (5).

## 3 Systems of Communicating Finite State Machines

Communicating Finite State Machines (CFSM) are a widely investigated automata-based formalism for the description and analysis of distributed systems, originally proposed in [11].

**Definition 3.1** (FSA)**.** *A* Finite State Automaton *is a quadruple* $(Q, q_0, \mathscr{L}, \delta)$ *where* $Q$ *is a finite set of states,* $q_0 \in Q$ *is the initial state,* $\mathscr{L}$ *a set of labels and* $\delta \subseteq Q \times \mathscr{L} \times Q$ *is a set of transitions.*
*An* $\varepsilon$-FSA *is an FSA such that* $\varepsilon \in \mathscr{L}$.

CFSMs are particular finite state automata representing processes which communicate by asynchronous exchanges of messages via FIFO channels. We now recall (partly following [12, 13, 17]) the definitions of CFSM and systems of CFSMs. We assume a countably infinite set $\mathbf{P}_{\mathbb{N}}$ of participant names (ranged over by $\mathtt{p}, \mathtt{q}, \mathtt{r}, \mathtt{h}, \mathtt{k}, \mathtt{v}, \mathtt{w} \ldots$) and a countably infinite alphabet $\mathbb{A}_{\mathbb{N}}$ of messages (ranged over by $\mathtt{a}$, $\mathtt{b}$, $\mathtt{c}$, $\mathtt{m}, \ldots$). We also assume an infinite set of indexes, ranged over by $i$, $j$, ... and use $I$, $J$, ... to range over finite sets of indexes.

**Definition 3.2** (Communicating Finite State Machine)**.** *Let* $\mathbf{P}$ *and* $\mathbb{A}$ *be finite subsets of* $\mathbf{P}_{\mathbb{N}}$ *and* $\mathbb{A}_{\mathbb{N}}$*, 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* $Act_{\mathbf{P},\mathbb{A}} = C_{\mathbf{P}} \times \{!, ?\} \times \mathbb{A}$
*Given an action* $l$ *we define* $\mathsf{sbj}(l)$ *(*the subject of *l) by* $\mathsf{sbj}(l) = \mathtt{p}$ *if* $l = \mathtt{pq}!\mathtt{m}$ *or* $l = \mathtt{qp}?\mathtt{m}$.

*iii) A* communicating finite state machine $M$ over $\mathbf{P}$ and $\mathbb{A}$ *is an FSA where* $\mathscr{L} = Act_{\mathbf{P},\mathbb{A}}$ *and 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. We assume $l, l', \ldots$ to range over actions and $w, w', \ldots$ to range over $\mathbb{A}^*_{\mathbb{N}}$ (the set of finite words over $\mathbb{A}_{\mathbb{N}}$). The symbol $\varepsilon\,(\notin \mathbb{A}_{\mathbb{N}})$ denotes the empty word and $\mid w \mid$ the length of a word $w$. The transitions of a CFSM are labelled by actions; a label $\mathtt{sr}!\mathtt{a}$ represents the asynchronous sending of message $\mathtt{a}$ from machine $\mathtt{s}$ to $\mathtt{r}$ through channel $\mathtt{sr}$ and, dually, $\mathtt{sr}?\mathtt{a}$ represents the reception (consumption) of $\mathtt{a}$ by $\mathtt{r}$ from channel $\mathtt{sr}$. 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* is a finite set of CFSMs referred to as *participants* of the communicating system.

**Definition 3.3** (Communicating system)**.** *Let* $\mathbf{P}$ *and* $\mathbb{A}$ *be as in Def. 3.2. 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* $\mathtt{p} \in \mathbf{P}$, $M_{\mathtt{p}} = (Q_{\mathtt{p}}, q_{0\mathtt{p}}, Act_{\mathbf{P},\mathbb{A}}, \delta_{\mathtt{p}})$ *is a CFSM over* $\mathbf{P}$ *and* $\mathbb{A}$.

The dynamics of a system are 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. A buffer is described as an element of $\mathbb{A}^*_{\mathbb{N}}$.

**Definition 3.4** (Configuration)**.** *Let* $S$ *be a communicating system over* $\mathbf{P}$ *and* $\mathbb{A}$. *A* configuration *of* $S$ *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 unread 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 will often denote a communicating system $(M_{\mathtt{p}})_{\mathtt{p}\in\{\mathtt{r}_i\}_{i\in I}}$ by $(M_{\mathtt{r}_i})_{i\in I}$.

Notice that the above definition of CFSM is generic with respect to the underlying sets $\mathbf{P}$ and $\mathbb{A}$. We shall often use $C$ and $Act$ for specific $C_{\mathbf{P}}$ and $Act_{\mathbf{P},\mathbb{A}}$ when no ambiguity can arise.

**Definition 3.5** (Transitions and reachable configurations)**.** *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 $\mathtt{a}\in\mathbb{A}$ such that one of the following conditions holds:*

1. $l=\mathtt{sr}!\mathtt{a}$ *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)* $w'_{\mathtt{sr}}=w_{\mathtt{sr}}\cdot\mathtt{a}$ *and for all* $\mathtt{pq}\neq\mathtt{sr}$ : $w'_{\mathtt{pq}}=w_{\mathtt{pq}}$;
2. $l=\mathtt{sr}?\mathtt{a}$ *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)* $w_{\mathtt{sr}}=\mathtt{a}\cdot w'_{\mathtt{sr}}$ *and for all* $\mathtt{pq}\neq\mathtt{sr}$ : $w'_{\mathtt{pq}}=w_{\mathtt{pq}}$.

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

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

The aim of analysing communication systems is to verify the properties that ensure certain pathological configurations, such as deadlock or reception errors, cannot be reached. We formalise now a number of relevant communication properties for systems of CFSMs that we deal with in the present paper.

**Definition 3.6** (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)* $\forall\mathtt{s}\in\mathbf{P}.[\ (q_{\mathtt{r}},\mathtt{sr}?\mathtt{a},q'_{\mathtt{r}})\in\delta_{\mathtt{r}}\ \Rightarrow\ (|w_{\mathtt{sr}}|>0\ \wedge\ w_{\mathtt{sr}}\notin\mathtt{a}\cdot\mathbb{A}^*)\ ]$.

*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 participant $\mathtt{r}$ 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 $s\to s'$ or* $\forall\mathtt{p}\in\mathbf{P}.\ q_{\mathtt{p}}$ *is final.*

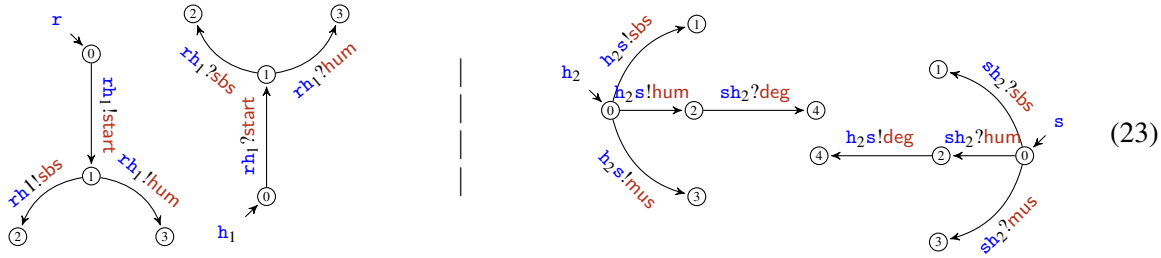
Note that the progress property (condition iv) implies deadlock freedom, but the converse is not true. For example, consider a non-final stuck configuration with a non-empty buffer. Communication properties above are essentially as presented in [11, 12, 13, 17].

# 4 PaI composition via partial gateways

In this section we formally define the PaI composition of CFSM systems via partial gateways. As mentioned before, for the sake of simplicity due of space motivations, we only consider the binary case. However, the definitions we provide naturally scale up to the composition of an arbitrary number of CFSM systems. To introduce and discuss the main issues, we use two communicating systems that implement the communications of the following systems as a running example.

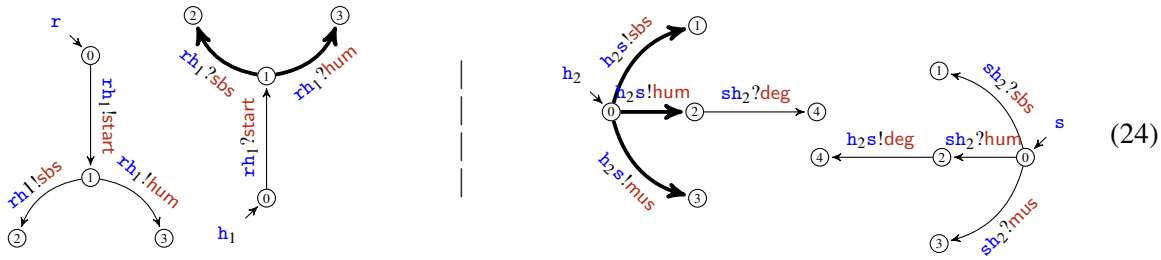
> Systems, $S_1$ and $S_2$ describe two simple home automation protocols. Participant $\mathtt{h}_1$ in $S_1$ is an autonomous robot that performs various housekeeping tasks once activated. It also functions as a humidifier. In order to perform this task, the robot remote controller $\mathtt{r}$ must send either the humidity parameter or the message sbs (set-by-sensor), indicating that the parameter is set by the robot's own sensor rule rather than the controller. System $S_2$, on the other hand, is a humidification system. Through the remote controller $\mathtt{h}_2$, one can instruct the controller $\mathtt{s}$ (through message hum) to compute the humidity parameter or to set it according to the humidifier's sensor (through message sbs). If the parameter is computed by the controller, the actual humidity level (deg) is sent back to the controller for display after a certain amount of time. Alternatively, one can send a parameter manually set by the user (mus).

For enhanced readability, all parts relating to our running example will be delimited by "►" and "◄".
► Focusing only on communications and abstracting from the actual values carried by the messages, the behaviour of the systems $S_1$ and $S_2$ previously described can be represented by communicating systems, as follows.



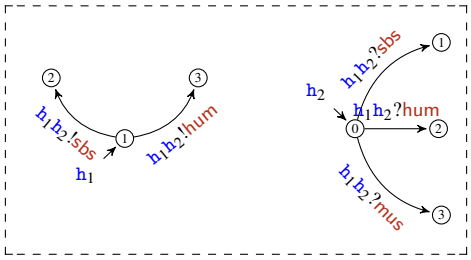

(23)

The standard PaI composition [8] of $S_1$ and $S_2$ using, respectively, $\mathtt{h}_1$ and $\mathtt{h}_2$ as interfaces, requires the interfaces to be replaced by forwarders, the gateways. In the present case, this replacement would be completely unreasonable, since it would mean to replace the entire robot with a gateway. A more reasonable approach would be to keep the robot as it is and use, for humidification, the more sophisticated humidification tool in $S_2$. To achieve this, rather than considering entire participants as interfaces, we can consider only some specific transitions as description of the behaviour of external systems. In the present example, for instance, we could still decide to compose $S_1$ and $S_2$ through $\mathtt{h}_1$ and $\mathtt{h}_2$. However, when interpreting the diagram (24) below from the perspective of $S_1$ and $S_2$, only the bold edges should be considered as actions belonging to an outer system.




(24)

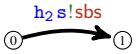
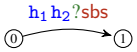
Intuitively, by considering the bold edges above as interface transitions expresses the intention to replace the robot's humidification task with the function of $S_2$. To achieve this, only messages relating to these transitions need to be forwarded. A possible connection policy should be then represented as a CFSM system that only deals with the interactions between the 'interface parts' of $\mathtt{h}_1$ and $\mathtt{h}_2$, as shown below.

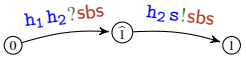


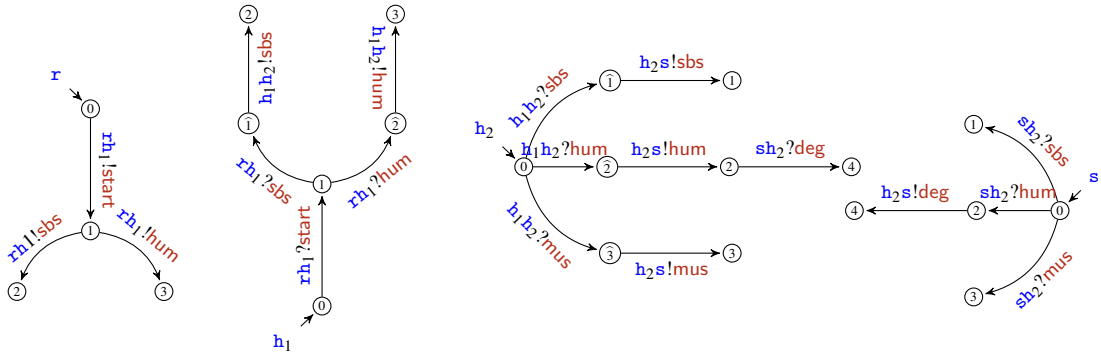

(25)

Such a communicating system enjoys all the communication properties of Definition 3.6, even if the $\mathtt{h}_1$ and $\mathtt{h}_2$ are not perfectly complementary, in particular even if state 3 of $\mathtt{h}_2$ cannot belong to any reachable configuration. ◀

In our envisaged approach, the composition would be achieved by replacing *participants with interface transitions* with *partial gateways* that forward only the interface parts' messages. The non-interface parts would continue to describe what the participants used for the connection are still in charge of. Each partial gateway is therefore constructed in a similar way to the standard composition via gateways, using the CFSMs of the interface participants and those forming the connection policy.

► In our example, out of the interface transition  of $\mathtt{h}_2$ in (24) and of the transition 

of $\mathtt{h}_2$ in the connection policy (25) , we introduce  in the resulting partial gateway, where $\widehat{1}$ is specifically introduced by the partial gateway construction. Note that, to enforce *conservativity* [8] – that is, to minimise the impact of the composition on the systems involved – partial gateways are given the same names as the corresponding participants with interface transitions. The above discussion applies, dually, for the interface transitions of $\mathtt{h}_1$ labelled with output actions. Non-interface transitions are instead left unchanged by the partial gateway construction. So, the resulting composed system in our running example is as follows.



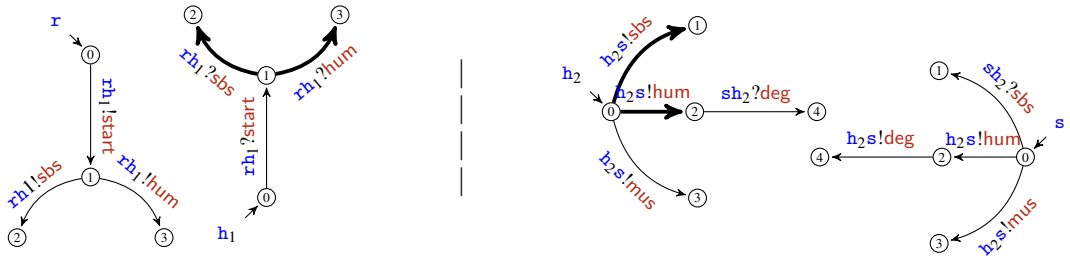

(26)

Our results guarantee that the above PaI composition via partial gateways satisfies the same communication properties (among those of Definition 3.6) satisfied by $S_1$ in $S_2$ in (23) and the connection policy in (25). ◀

However, in general, once two interface participants have been chosen, some care must be taken when deciding which of their transitions to consider as interface transitions. Not all transitions are suitable for

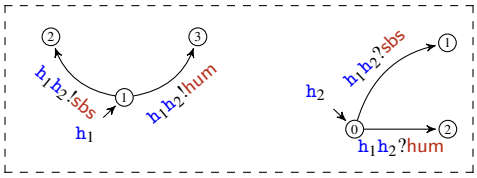
this purpose. That is why in Definition 4.2 below we define the set of the *interface decorations* of a CFSM, where an interface decoration corresponds to a possible suitable selection of interface transitions preventing issues like those we discuss below.

► Consider again $\mathtt{h}_1$ and $\mathtt{h}_2$ of our running example, in particular the following interface transitions.

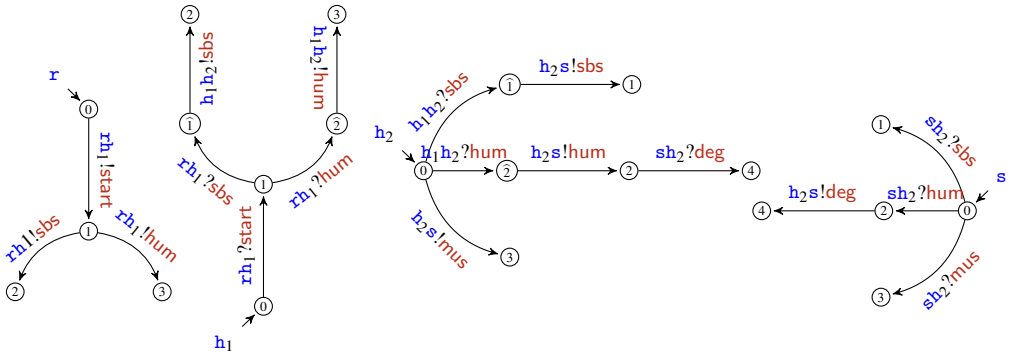



(27)

The corresponding connection policy would now be



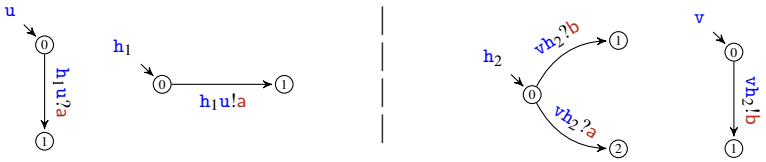

This communicating system, as well as $S_1$ and $S_2$, is orphan-message free. The composition obtained by building partial gateways out of such connection policy is then




(28)

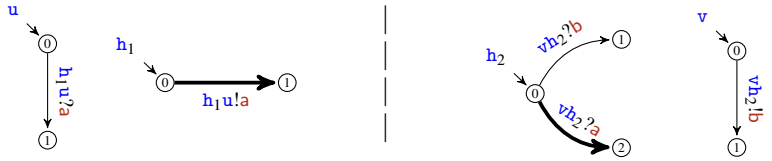
This communicating system, however, is not orphan-message free. In fact the following configuration, made by final states only, is reachable: $s = ((3_{\mathtt{r}}, 3_{\mathtt{h}_1}, 3_{\mathtt{h}_2}, 3_{\mathtt{s}}), \vec{w})$, where $\vec{w} \neq \vec{\varepsilon}$, in particular $w_{\mathtt{h}_1\mathtt{h}_2} = \langle \mathsf{hum} \rangle$. Hence $s$ is an orphan-message configuration. ◄

The problem in the above discussion arises from the fact that the resulting partial gateway $\mathtt{h}_2$ is actually a CFSM with mixed states. However, issues can arise also in the absence of mixed states. Consider the two following systems $S_1$ and $S_2$ we wish to compose through, $\mathtt{h}_1$ and $\mathtt{h}_2$. Both the systems satisfy the progress property.



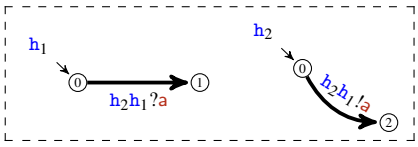

(29)

We then make the following choice of interface transitions for the interface participants.

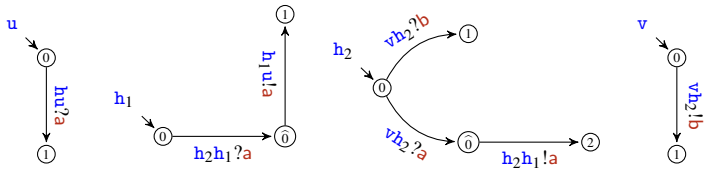



(30)

The following (unique) connection policy satisfies the progress property.

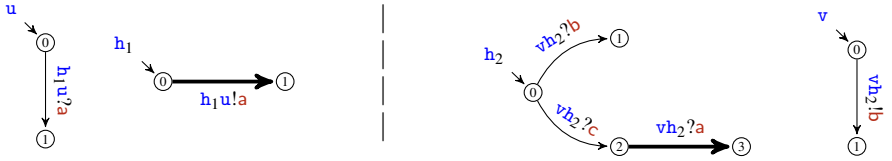



However, the PaI composition via partial gateway below does not satisfies the progress property.



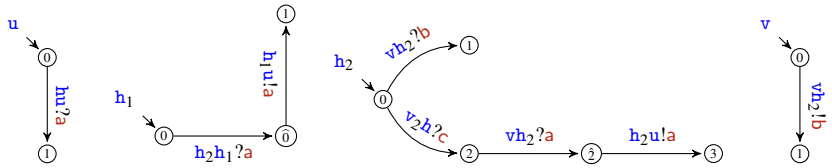

In fact, some states in $s = (0_{\mathtt{u}}, 0_{\mathtt{h}_1}, 1_{\mathtt{h}_2}, 1_{\mathtt{v}})$ are not final, $s$ is reachable and $s \not\longrightarrow$.

In both of the previous examples, there is a state that contains both normal outgoing transitions and outgoing interface transitions. However, the following example shows that progress preservation can also be disrupted by states with no outgoing interface transitions.




(31)

The (unique) connection policy satisfies progress. The composition below, however, does not.




In fact, some states in the configuration $s = (0_{\mathtt{u}}, 0_{\mathtt{h}_1}, 1_{\mathtt{h}_2}, 1_{\mathtt{v}})$ are not final, $s$ is reachable and $s \not\longrightarrow$. Intuitively, the main problem with some of the previous examples is that the choice of transition in the composition does not depend on a single participant anymore, involving actually both $\mathtt{h}_1$ and $\mathtt{h}_2$. In the last example, problems also arise when a choice that depends on $\mathtt{h}_2$ conflicts with what the other partial gateways does. We therefore need to avoid such issues, possibly by syntactic means. Our proposal requires outgoing non-interface transitions must always be single, i.e. if a state has an outgoing non-interface transition, this must be its only outgoing transition. We formalise such restriction in the following definition of CFSM with interface transitions, where the symbol $\iota$ is intended to identify interface transitions and $\not\iota$ is intended to identify non-interface transitions.

**Definition 4.1** (CFSM with interface transitions)**.** *A* CFSM with interface transitions *(CFSM$^{\mathsf{it}}$ for short) is a tuple $M = (Q, q_0, Act, \boldsymbol{\delta})$ where $Q$, $q_0$ and $Act$ are as in the definition of CFSM, whereas*

$$\boldsymbol{\delta} \subseteq Q \times Act \times Q \times \{\iota, \not\iota\}$$

*and such that* - $(q_1,l,q_2,x),(q_1,l,q_2,y) \in \boldsymbol{\delta} \Rightarrow x = y$.
- $(q_1,l,q_2,\not\iota),(q_1,l',q_2',z) \in \boldsymbol{\delta} \Rightarrow (l = l' \text{ and } q_2 = q_2')$
*(and hence $z = \not\iota$ by the previous item).*

*An element of $\boldsymbol{\delta}$ of the form $(\_,\_,\_,\iota)$ is called* interface transition.

As done in the previous examples, we use the notation $q \xrightarrow{l} q'$ for $(q,l,q',\not\iota)$ and $q \xrightarrow{l} q'$ for $(q,l,q',\iota)$.

It is possible to check that $\mathtt{h}_2$ in (24) above is a CFSM with interface transitions, whereas participants $\mathtt{h}_2$ in (27), (30) and (31) are not. It is worth also noting that the usual PaI composition via gateway is a particular case of composition via partial gateways where all transitions are interface transitions. Absence of non interface transition makes conditions in Definition 4.1 above vacuously satisfied.

We now describe the various stages of the PaI composition via partial gateways that have been roughly presented so far, and provide formal definitions of the involved notions. We continue to use our running example to illustrate the various stages.

**Indentifying interface participants and selecting interface transitions.** PaI composition via partial gateways consists, first of all, in identifying one participant per system and then properly selecting some transitions in order to get CFSMs with interface transitions. Given a CFSM $M$, an interface decoration is one of the possible CFSM with interface transitions that we can get out of $M$.

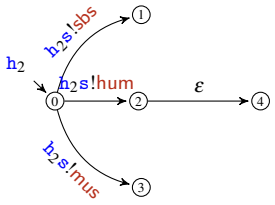
**Definition 4.2** (Interface decorations)**.** *Let $M = (Q,q_0,Act,\delta)$ be a CFSM. We define the* interface decoration set *of $M$ as the following set of CFSM$^{\mathsf{it}}$s:*

$$\mathsf{IDS}(M) = \{(Q,q_0,Act,\boldsymbol{\delta}) \mid (Q,q_0,Act,\boldsymbol{\delta}) \text{ is a CFSM}^{\mathsf{it}} \text{ with } \boldsymbol{\delta}\restriction_{Q\times Act\times Q} = \delta\}$$

*where $\boldsymbol{\delta}\restriction_{Q\times Act\times Q} = \{(q,l,q') \mid \exists x \; s.t. (q,l,q',x) \in \boldsymbol{\delta}\}$.*
*We refer to an element of $\mathsf{IDS}(M)$ as* an interface decoration of $M$.

▶ By the above definition, $\mathtt{h}_1$ and $\mathtt{h}_2$ in (24) are interface decorations for $\mathtt{h}_1$ and $\mathtt{h}_2$ in (23). In the following we focus for simplicity just on the interface of $S_2$. ◀

**Extract the intended interface behaviours from the chosen decorations.** Given a CFSM with interface transitions – for example $\mathtt{h}_2$ in (24) – these transitions identify the behaviour of the external system that we intend to connect to via the composition. Such a behaviour is represented by focusing on interface transitions only, disregarding the non-interface ones, which we label by $\varepsilon$.
▶ From $\mathtt{h}_2$ in (24) we hence intend to get,




(32)

◀

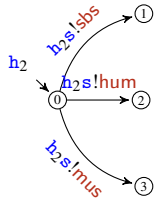
We use the $\varepsilon$ symbol since we manipulate it as the empty string in automata theory.
To formalise the process through which we get (32) from the $\mathtt{h}_2$ of (24), we use the following function.

**Definition 4.3** ($\varepsilon(M)$)**.**

*Let $M = (Q,q_0,Act,\boldsymbol{\delta})$ be a CFSM$^{\mathsf{it}}$. We define $\varepsilon(M)$ as the $\varepsilon$-FSA $(Q,q_0,Act\cup\{\varepsilon\},\delta')$ where*

$$\delta' = \{(q,\varepsilon,q') \mid (q,l,q',\not\iota) \in \boldsymbol{\delta}\} \cup \{(q,l,q') \mid (q,l,q',\iota) \in \boldsymbol{\delta}\}$$

From an $\varepsilon$-FSA like (32) we need now to get a proper CFSM which behaves the same[1].
▶ In our running example such a behaviour should hence correspond to the following CFSM.




(33)

◀

According to the conditions in Definition 4.1, outgoing transitions in a CFSM$^{\mathsf{it}}$ are single if they are non-interface transitions. Therefore, the $\varepsilon$ transitions in the CFSM obtained by the application of $\varepsilon(\_)$ are also single. With this restriction, the interface behaviour can be obtained by simply making all the $\varepsilon$-transition "collapse". This transformation is a much simpler algorithm than the standard algorithms for eliminating $\varepsilon$-transitions in $\varepsilon$-FSA, due to our strong restriction on $\varepsilon$-transitions. In general, however, the CFSM that describes the behaviour of the $\varepsilon$-FSA might even differ from that obtained using the aforementioned algorithm, provided it is possible to establish a precise correspondence between the non-$\varepsilon$ transitions in the $\varepsilon$-FSA and the CFSM transitions. We therefore introduce the following definition which focuses not on a particular algorithm for eliminating $\varepsilon$-transitions, but rather on the properties that the result must satisfy.

**Definition 4.4** ($\not\varepsilon$-versions). *Let $M = (Q, q_0, Act \cup \{\varepsilon\}, \delta)$ be a $\varepsilon$-FSA and $M' = (Q', q_0', Act, \delta)$ be a CFSM. We say that $M'$ is a* $\not\varepsilon$-version of $M$ via $f$ *($\not\varepsilon_f$-version of $M$ for short) whenever*

*a) $M'$ is deterministic;*

*b) $f : Q \to Q'$ is onto and such that $f(q_0) = f(q_0')$;*

*c) $q_1 \xrightarrow{\varepsilon} q_2$ implies $f(q_1) = f(q_2)$;*

*d) $q_1 \xrightarrow{l} q_2$ implies $f(q_1) \xrightarrow{l} f(q_2)$;*

*e) $f(q_1) \xrightarrow{l} f(q_2)$ implies $q_1 \stackrel{l}{\Rightarrow} q_2$ (i.e. $q_1 \xrightarrow{\varepsilon}{}^* q_1' \xrightarrow{l} q_2' \xrightarrow{\varepsilon}{}^* q_2$ for some $q_1'$ and $q_2'$).*

▶ The CFSM in (33 is a $\not\varepsilon_f$-version of the $\varepsilon$-FSA in (32) where

$$f : \{0,1,2,3,4\} \to \{0,1,2,3\} \text{ with } f(4) = 2 \text{ and } f(q) = q \text{ for } q \in \{0,1,2,3\}. \tag{34}$$

◀

**Describing the forwarding behaviours i.e. the connection policy.** As previously mentioned, a connection policy is a communicating system that describes how in the intended composition, partial gateways forward the messages involved in interface transitions. We can obtain an element for the desired connection policy from a $\not\varepsilon$-version of an $\varepsilon$-FSA by $(a)$ "dualising" the actions labelling the transitions and $(b)$ replacing the senders in inputs and the receivers in outputs with participant names of the connection policy. Note that, unlike in the composition of multiple systems, such a replacement is uniquely determined in the binary composition case. In the following definition of **P**-duality, **P** is the set of the names of the other participants in the intended connection policy.

**Definition 4.5** (**P**-duality). *i) Let $l, l' \in Act$ and let $\mathbf{P} \neq \emptyset$ be a set of participants. We say that $l'$ is* a **P**-dual of $l$ *whenever*

[1] The usual algorithm for $\varepsilon$-FSA are not useful for our purposes.

- $l = \mathtt{rq}?\mathsf{m} \;\Rightarrow\; l' = \mathtt{qs}!\mathsf{m}$ *with* $\mathtt{s} \in \mathbf{P}$;
- $l = \mathtt{qr}!\mathsf{m} \;\Rightarrow\; l' = \mathtt{sq}?\mathsf{m}$ *with* $\mathtt{s} \in \mathbf{P}$;

*ii) Let* $\delta, \delta' \in Q \times Act \times Q$ *and let* $\mathbf{P}$ *be a set of participants. We say that* $\delta'$ *is a* $\mathbf{P}$-dual of $\delta$ *whenever* $\delta'$ *is a minimal relation over* $Q \times Act \times Q$ *such that*

$$q \xrightarrow{l} q' \in \delta \;\; \textit{implies} \;\; q \xrightarrow{l'} q' \in \delta', \textit{ where } l' \textit{ is a } \mathbf{P}\textit{-dual of } l.$$

*iii) Let* $M = (Q, q_0, Act, \delta)$ *and* $M' = (Q, q_0, Act, \delta')$ *be two CFSMs and let* $\mathbf{P}$ *be a set of participants. We say that* $M'$ *is a* $\mathbf{P}$-dual of $M$ *whenever* $\delta'$ *is a* $\mathbf{P}$*-dual of* $\delta$.

▶ We have that $\mathtt{h}_2$ in (25) is a $\{\mathtt{h}_1\}$-dual of (33) (actually the only $\{\mathtt{h}_1\}$-dual, since $\{\mathtt{h}_1\}$ is a singleton by the fact that in our running example we consider binary composition). ◀

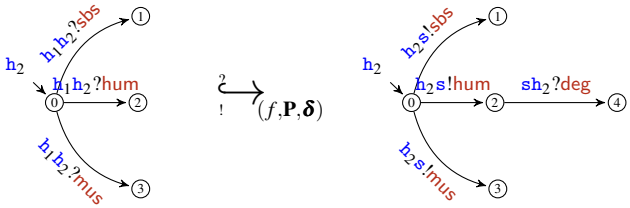
In order to formalise the notion of connection policy, we now define a binary relation on CFSMs. Two CFSMs are related if the first CFSM is an element of a possible connection policy, and the second CFSM is a participant whose certain transitions are intended to serve as interface transitions. This relation depends on the concepts of interface decoration, $\varepsilon(\_)$, $\not\boldsymbol{\varepsilon}$-version and $\mathbf{P}$-duality.

**Definition 4.6** ($\mathbf{P}$-embeddability)**.** *Let* $M^1_{\mathtt{h}} = (Q, q_0, Act, \delta^1)$ *and* $M^2_{\mathtt{h}} = (Q, q_0, Act, \delta^2)$ *be two CFSMs with the same name* $\mathtt{h}$*, and let* $\mathbf{P}$ *be a set of participants. Moreover, let* $\boldsymbol{\delta} \subseteq Q \times Act \times Q \times \{\imath, \not\imath\}$.
*We say that* $M^1_{\mathtt{h}}$ *is* $\mathbf{P}$-embeddable into $M^2_{\mathtt{h}}$ via $\boldsymbol{\delta}$ and $f$, *written* $M^1_{\mathtt{h}} \overset{?}{!}\!\!\rightarrow_{(f,\mathbf{P},\boldsymbol{\delta})} M^2_{\mathtt{h}}$, *whenever*

$$M^1_{\mathtt{h}} \textit{ is a } \mathbf{P}\textit{-dual of a } \not\boldsymbol{\varepsilon}_f\textit{-version of } \varepsilon(M'_{\mathtt{h}})$$

*where* $M'_{\mathtt{h}} = (Q, q_0, Act, \boldsymbol{\delta}) \in \mathsf{IDS}(M^2_{\mathtt{h}})$. *We refer to* $\varepsilon(M'_{\mathtt{h}})$ *as the* $\varepsilon$-counterpart of $M^1_{\mathtt{h}}$. *We write simply* $M^1_{\mathtt{h}} \overset{?}{!}\!\!\rightarrow M^2_{\mathtt{h}}$ *whenever there exist* $\mathbf{P}$, $\boldsymbol{\delta}$ *and* $f$ *such that* $M^1_{\mathtt{h}} \overset{?}{!}\!\!\rightarrow_{(f,\mathbf{P},\boldsymbol{\delta})} M^2_{\mathtt{h}}$ *and in this case we say that* $M^1_{\mathtt{h}}$ is embeddable into $M^2_{\mathtt{h}}$.

▶ In our running example we have that




(35)

where $f$ is as in (34), $\mathbf{P} = \{\mathtt{h}_1\}$ and $\boldsymbol{\delta}$ is as for $\mathtt{h}_2$ in (24). ◀

We can now formally define the notion of connection policy, intended as the description of the way partial gateways has to communicate in a PaI composition.

**Definition 4.7** (Connection policy)**.** *Let* $S_i = (M^i_{\mathtt{x}})_{\mathtt{x} \in \mathbf{P}_i}$ *be two communicating systems with interface participants* $\mathtt{h}_i$ *(*$i = 1, 2$*). A* connection policy *for the set of interface participants* $H = \{\mathtt{h}_1, \mathtt{h}_2\}$ *is a communicating system* $\mathbb{C} = (M^{\mathbb{C}}_{\mathtt{u}})_{\mathtt{u} \in H}$ *such that, for each* $i \in \{1, 2\}$, *for some* $f_i$ *and* $\boldsymbol{\delta_i}$: $M^{\mathbb{C}}_{\mathtt{h}_i} \overset{?}{!}\!\!\rightarrow_{(f_i, H \setminus \{\mathtt{h}_i\}, \boldsymbol{\delta_i})} M^i_{\mathtt{h}_i}$.

▶ The communicating system of (25) is hence a connection policy for our running example. ◀

**Build the partial gateways.** In the PaI composition via partial gateways, such gateways are constructed using the interface participants and their corresponding participants in a connection policy (those possessing the same name).

**Definition 4.8** (Partial Gateway)**.** *Let* $M^1_{\mathtt{h}} = (Q_1, q^1_0, Act, \delta^1)$ *and* $M^2_{\mathtt{h}} = (Q_2, q^2_0, Act, \delta^2)$ *such that* $M^1_{\mathtt{h}} \overset{?}{\underset{!}{\leftrightarrow}}_{(f,\mathbf{P},\boldsymbol{\delta})} M^2_{\mathtt{h}}$*. The* partial gateway $M^1_{\mathtt{h}} \downdownarrows M^2_{\mathtt{h}}$ *obtained from* $M^1_{\mathtt{h}}$ *and* $M^2_{\mathtt{h}}$ *is the CFSM with name* $\mathtt{h}$ *defined by*

$$M^1_{\mathtt{h}} \downdownarrows M^2_{\mathtt{h}} = (Q^2 \cup \widehat{Q}, q^2_0, Act, \widehat{\delta})$$

*where*
- $\widehat{Q} = \bigcup_{q \in Q} \{q^{(q,l,q')} \mid (q,l,q',\iota) \in \boldsymbol{\delta}\}$;
- $\widehat{\delta} = \{(q, \mathtt{rh?a}, \widehat{q}), (\widehat{q}, \mathtt{hs!a}, q') \mid (q, \mathtt{hs!a}, q', \iota) \in \boldsymbol{\delta}, (f(q), \mathtt{rh?a},\ f(q')) \in \delta^1,\ \widehat{q} = q^{(q,\mathtt{hs!a},q')}\} \cup$
  $\{(q, \mathtt{sh?a}, \widehat{q}), (\widehat{q}, \mathtt{hr!a}, q') \mid (q, \mathtt{sh?a}, q', \iota) \in \boldsymbol{\delta},\ (f(q), \mathtt{hr!a}, f(q')) \in \delta^1,\ \widehat{q} = q^{(q,\mathtt{sh?a},q')}\} \cup$
  $\{(q,l,q') \mid (q,l,q',\not\iota) \in \boldsymbol{\delta}\}$.

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

► Using the two participants $\mathtt{h}_2$ in (35) we can build the partial gateway intended to be substituted for $\mathtt{h}_2$ in the composition. Namely $\mathtt{h}_2$ in (26). ◄

A simple condition has to be satisfied in order two communicating systems can be elegible for PaI composition via partial gateways.

**Definition 4.9** (Composability)**.** *Let* $S_1$ *and* $S_2$ *be two communicating systems such that* $S_i = (M^i_{\mathtt{x}})_{\mathtt{x} \in \mathbf{P}_i}$ *(*$i = 1,2$*). We say that* $S_1$ *and* $S_2$ *are* composable *whenever* $\mathbf{P}_1 \cap \mathbf{P}_2 = \emptyset$*.*

**Definition 4.10** (PaI composition of communicating systems via partial gateways)**.** *Let* $S_1$ *and* $S_2$ *be two composable communicating systems such that* $S_i = (M^i_{\mathtt{x}})_{\mathtt{x} \in \mathbf{P}_i}$ *(*$i \in \{1,2\}$*) and let* $\mathbb{C} = (M^{\mathbb{C}}_{\mathtt{u}})_{\mathtt{u} \in H}$ *be a connection policy for the interface participants* $H = \{\mathtt{h}_1, \mathtt{h}_2\}$*. The* PaI composition via partial gateways of $S_1$ and $S_2$ with respect to $\mathbb{C}$ *is the communicating system*

$$\mathscr{PC}(\{S_i\}_{i \in \{1,2\}}, \mathbb{C}) = (M'_{\mathtt{p}})_{\mathtt{p} \in \mathbf{P}_1 \cup \mathbf{P}_2}$$

*where*

$$M'_{\mathtt{p}} = \begin{cases} M^i_{\mathtt{p}} & \text{if } \mathtt{p} \notin H \text{ and } \mathtt{p} \in \mathbf{P}_i \\ M^{\mathbb{C}}_{\mathtt{h}_i} \downdownarrows M^i_{\mathtt{h}_i} & \text{if } \mathtt{p} = \mathtt{h}_i \text{ and } i \in \{1,2\} \end{cases}$$

We claim that all the previous definitions can be extended in a natural way for the composition of an arbitrary number of systems where in each of them an interface participant is selected.

For most properties, preservation by composition requires the no-mixed-state condition (see the notations after Definition 3.2) to be met. Indeed, the presence of mixed states among the interface participants would otherwise result in some counterexamples that can be obtained by adapting those in [8, 9].

**Theorem 4.11** (Safety of PaI composition via partial gateways)**.** *Let* $S_1$ *and* $S_2$ *be two composable communicating systems with, respectively, interface participants* $\mathtt{h}_1$ *and* $\mathtt{h}_2$ *with no mixed states; and let* $\mathbb{C}$ *be a connection policy for* $H = \{\mathtt{h}_1, \mathtt{h}_2\}$*.*

*Let* $\mathscr{P}$ *be either the property of* deadlock-freedom *or* reception-error-freedom *or* progress *(as defined in Definition 3.6). If* $\mathscr{P}$ *holds for both* $S_1$ *and* $S_2$*, as well as for* $\mathbb{C}$*, then* $\mathscr{P}$ *holds for* $\mathscr{PC}(\{S_i\}_{i \in \{1,2\}}, \mathbb{C})$*. Moreover, the above holds also if the no-mixed-state condition is removed and* $\mathscr{P}$ *is* orphan-message-freedom*.*

The proof of the above theorem descends from the fact that any system obtained by PaI composition via partial gateways can be actually got by two applications of (binary) partial-fusion composition (Proposition 5.4) and that this particular composition is property preserving (Theorem 5.3). In the following section we define partial-fusion composition and provide the mentioned results.

# 5 Partial-fusion Composition

We consider now a particular composition of two communicating systems where a participant of a system is embeddable in a participant of the other system. Their *partial-fusion* composition is obtained by replacing the two participants with a single participant obtained throught the gateway construction of Definition 4.8 that in the present context we call *partial fusion*.

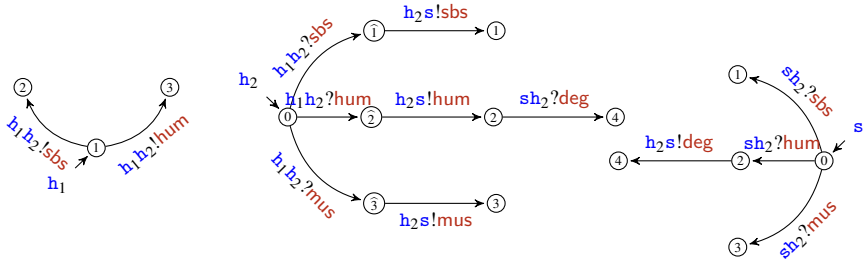
**Definition 5.1** (Composition by Partial Fusion)**.** *Let* $S_1 = (M^1_{\mathtt{x}})_{\mathtt{x}\in\mathbf{P}_1}$ *and* $S_2 = (M^2_{\mathtt{x}})_{\mathtt{x}\in\mathbf{P}_2}$ *be two communicating systems such that* $\mathbf{P}_1 \cap \mathbf{P}_2 = \{\mathtt{h}\}$ *and* $M^1_{\mathtt{h}} \overset{?}{\underset{!}{\mapsto}}_{(f,\boldsymbol{\delta},\mathbf{P}_1\setminus\{\mathtt{h}\})} M^2_{\mathtt{h}}$*. We define the* composition of $S_1$ and $S_2$ via partial fusion of $\mathtt{h}$ *by*

$$\mathscr{FC}_{\mathtt{h}}(S_1,S_2) = (\widetilde{M}_{\mathtt{x}})_{\mathtt{x}\in\mathbf{P}_1\cup\mathbf{P}_2}$$

*where*
- $\widetilde{M}_{\mathtt{x}} = M^1_{\mathtt{x}}$ *if* $\mathtt{x}\in\mathbf{P}_1\setminus\{\mathtt{h}\}$;
- $\widetilde{M}_{\mathtt{x}} = M^2_{\mathtt{x}}$ *if* $\mathtt{x}\in\mathbf{P}_2\setminus\{\mathtt{h}\}$;
- $\widetilde{M}_{\mathtt{h}} = M^1_{\mathtt{h}} \between M^2_{\mathtt{h}}$

*We refer to* $\widetilde{M}_{\mathtt{h}}$ *as the* connector *of* $S_1$ *and* $S_2$ *in the composition. We say the connector* $\widetilde{M}_{\mathtt{h}}$ *is obtained by* partial fusion *of* $M^1_{\mathtt{h}}$ *and* $M^2_{\mathtt{h}}$.

**Example 5.2.** By taking as $S_1$ the system in (25) and as $S_2$ the system on the right in (23) we can check, as done in (35), that $M^1_{\mathtt{h}_2}$ is embeddable into $M^2_{\mathtt{h}_2}$. Hence, the composition of $S_1$ and $S_2$ via partial fusion of $\mathtt{h}_2$ returns the system




◇

**Theorem 5.3** (Safety of partial fusion composition)**.** *Let* $S_1 = (M^1_{\mathtt{x}})_{\mathtt{x}\in\mathbf{P}_1}$ *and* $S_2 = (M^2_{\mathtt{x}})_{\mathtt{x}\in\mathbf{P}_2}$ *be two communicating systems such that* $\mathbf{P}_1 \cap \mathbf{P}_2 = \{\mathtt{h}\}$ *and where* $M^1_{\mathtt{h}}$ *has no mixed state. Moreover, let* $M^1_{\mathtt{h}} \overset{?}{\underset{!}{\mapsto}}_{(f,\boldsymbol{\delta},\mathbf{P}_1\setminus\{\mathtt{h}\})} M^2_{\mathtt{h}}$*. Let now* $\mathscr{P}$ *be either the property of* deadlock-freedom *or* reception-error-freedom *or* progress *(as defined in Definition 3.6). If* $\mathscr{P}$ *holds for both* $S_1$ *and* $S_2$*, then* $\mathscr{P}$ *holds for* $\mathscr{FC}_{\mathtt{h}}(S_1,S_2)$*. Moreover, the above holds also if the no-mixed-state condition is removed and* $\mathscr{P}$ *is* orphan-message-freedom.

The theorem above can be proved by contradiction. Specifically, one shows that if $\mathscr{P}$ fails to hold for $S$, then it must already fail for either $S_1$ or $S_2$. Following the approach of [8, 9], a central concept is that of *projection* of a reachable configuration of the composed system onto the corresponding configurations of the individual systems $S_1$ and $S_2$. In fact, one can prove that the projections of reachable configurations that do not involve intermediate gateway states are themselves reachable configurations. The proof of Theorem 5.3 can be found in [1].

Theorem 4.11 can be obtained as a corollary of Theorem 5.3 above by using the following proposition, which states that any binary PaI-composition via partial gateways can actually be obtained by applying the partial-fusion composition twice. First between one of the systems to be composed and

the connection policy; then beween the other system and what has been obtained by the first fusion-composition. This result follows from the definitions of PaI composition via partial gateway and partial-fusion composition.

**Proposition 5.4.** *Let $S_1$ and $S_2$ be two composable communicating systems such that $S_i = (M^i_{\mathtt{x}})_{\mathtt{x}\in\mathbf{P}_i}$ ($i \in \{1,2\}$) and let $\mathbb{C} = (M^{\mathbb{C}}_{\mathtt{u}})_{\mathtt{u}\in H}$ be a connection policy for the interface participants $H = \{\mathtt{h}_1, \mathtt{h}_2\}$.*

$$\mathscr{PC}(\{S_i\}_{i\in\{1,2\}}, \mathbb{C}) = \mathscr{FC}_{\mathtt{h}_1}(S_1, \mathscr{FC}_{\mathtt{h}_2}(\mathbb{C}, S_2))$$

# 6 Conclusions

The "PaI composition via partial gateway" approach is a method of composing two systems. This approach transforms one participant from each system into a partial gateway. Any two participants can be chosen. The partial gateways are obtained from the chosen participants and a connection policy. One can choose one's preferred connection policy as long as certain requirements (such as embeddability) are met. The resulting composite system is guaranteed to satisfy many properties that are satisfied by all the individual systems and the connection policy. The way in which such properties are proven for the individual systems or the connection policy is beyond the scope of the present paper. The property preservation guaranteed by this method of system composition is shown to descend from a similar result for a simpler method of system composition: composition by fusion.

A thorough discussion on works related to system composition in general and PaI composition in particular, can be found in [9, Sect.8]. Although numerous approaches to system composition in the literature can support some form of partial composition – for example through selective interfaces [15], partial specifications [16], or modular synchronisation in process algebras like CCS or CSP – the PaI approach via partial gateways looks conceptually distinct to the best of our knowledge.

We are planning to formally define partial versions of the PaI multicomposition and orchestrated multicomposition [8, 9] and to show that they can be obtained by appropriate numbers of application of the binary fusion-composition proposed in this paper. The property preservation of those composition approaches would then be entailed by the property preservation result of partial fusion composition. It would also be interesting to investigate whether and how our approach fits into the *composition calculus* [14], a very abstract framework of modules and their composition.

Our results does not apply to the property of lock-freedom, i.e. the property guaranteeing the absence of reachable configurations where some participant remains stuck in all possible continuations (see [8, 9] for a formal definition). Actually, counterexamples for lock-freedom preservation can be obtained by simply adapting [8, Example 5.7] for both PaI composition via partial gateways and partial fusion. We anticipate that the simple and straightforward nature of partial fusion will enable us to focus on the essential issues underlying the lack of lock-freedom preservation, and hence identify appropriate and manageable conditions ensuring it.

**Acknowledgements** We would like to express our gratitude to the anonymous referees for their careful reading of the paper, their helpful comments and suggestions, and their valuable interactions in the ICE discussion forum. We would also like to thank Emilio Tuosto for providing us with macros to draw automata. Special thanks are also due to Mariangiola Dezani and Rolf Hennicker for their support and friendship.

# References


[1] Franco Barbanera: *Safe Composition of CFSM Systems via Partial Gateways (extended)*. Available at `https://github.com/francobarbanera/CFSM-partial-fusion-extended/blob/main/CFSM-partial-fusion-ext.pdf`.

[2] Franco Barbanera, Viviana Bono & Mariangiola Dezani-Ciancaglini (2025): *Open compliance in multiparty sessions with partial typing*. *Journal of Logical and Algebraic Methods in Programming* 144, p. 101046, doi:https://doi.org/10.1016/j.jlamp.2025.101046.

[3] Franco Barbanera, Ugo de'Liguoro & Rolf Hennicker (2018): *Global Types for Open Systems*. In: Proceedings ICE 2018, *Electronic Proceedings in Theoretical Computer Science* 279, Open Publishing Association, pp. 4–20, doi:10.4204/EPTCS.279.4.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] Franco Barbanera & Rolf Hennicker (2024): *Safe Composition of Systems of Communicating Finite State Machines*. In Clément Aubert, Cinzia Di Giusto, Simon Fowler & Violet Ka I Pun, editors: Proceedings 17th *Interaction and Concurrency Experience*, Groningen, The Netherlands, 21st June 2024, *Electronic Proceedings in Theoretical Computer Science* 414, Open Publishing Association, pp. 39–57, doi:10.4204/EPTCS.414.3.

[9] Franco Barbanera & Rolf Hennicker (2026): *Safe Orchestrated Multicomposition of Systems of Communicating Finite State Machines*. *Journal of Logical and Algebraic Methods in Programming*, p. 101109, doi:https://doi.org/10.1016/j.jlamp.2026.101109. Available at `https://www.sciencedirect.com/science/article/pii/S2352220826000015`.

[10] 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.

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

[12] 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.

[13] 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.

[14] Peter Fettke & Wolfgang Reisig (2024): *Once and for All: How to Compose Modules - The Composition Calculus*. In Tiziana Margaria & Bernhard Steffen, editors: *Leveraging Applications of Formal Methods, Verification and Validation. Rigorous Engineering of Collective Adaptive Systems - 12th International Symposium, ISoLA 2024, Crete, Greece, October 27-31, 2024, Proceedings, Part II*, *Lecture Notes in Computer Science* 15220, Springer, pp. 173–190, doi:10.1007/978-3-031-75107-3_11.

[15] Thomas A. Henzinger (2003): *Automata for Specifying Component Interfaces*. In Oscar H. Ibarra & Zhe Dang, editors: *Implementation and Application of Automata, 8th International Conference, CIAA 2003, Santa Barbara, California, USA, July 16-18, 2003, Proceedings*, *Lecture Notes in Computer Science* 2759, Springer, pp. 1–2, doi:10.1007/3-540-45089-0_1.

[16] Ramon Janssen (2020): *Combining Partial Specifications using Alternating Interface Automata*. In Heike Wehrheim & Jordi Cabot, editors: *Fundamental Approaches to Software Engineering - 23rd International Conference, FASE 2020, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2020, Dublin, Ireland, April 25-30, 2020, Proceedings*, *Lecture Notes in Computer Science* 12076, Springer, pp. 462–481, doi:10.1007/978-3-030-45234-6_23.

[17] 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.