Composing Communicating Systems, Synchronously

Composing Communicating Systems, Synchronously*

Franco Barbanera[1], Ivan Lanese[2], Emilio Tuosto[3]

[1] Dept. of Mathematics and Computer Science, University of Catania (Italy)
[2] Focus Team, University of Bologna/INRIA (Italy)
[3] Gran Sasso Science Institute (Italy) and University of Leicester (UK)

Abstract. Communicating systems are nowadays part of everyday life, yet programming and analysing them is difficult. One of the many reasons for this difficulty is their size, hence compositional approaches are a need. We discuss how to ensure relevant communication properties such as deadlock freedom in a compositional way. The idea is that communicating systems can be composed by taking two of their participants and transforming them into coupled forwarders connecting the two systems. It has been shown that, for asynchronous communications, if the participants are "compatible" then composition satisfies relevant communication properties provided that the single systems satisfy them. We show that such a result changes considerably for synchronous communications. We also discuss a different form of composition, where a unique forwarder is used.

1 Introduction

The behaviour of systems which communicate via point-to-point message passing can be described in terms of systems of Communicating Finite State Machines (CFSMs) [10], that is systems of finite state automata whose transitions are labelled by sending and receiving actions. Such systems can be then analysed to check whether they enjoy relevant communication properties such as deadlock freedom, lock freedom, etc. (see, e.g., [16,24,7,20,6]).

Traditionally these systems are viewed as *closed*, thus one needs full knowledge of the whole system in order to analyse it. In scenarios such as the Internet, the Cloud or serverless computing, such assumption is less and less realistic.

Recently, an approach to the composition of systems of CFSMs has been proposed [3,4]. The main idea of the approach is to take two systems, select two of their participants (one per system) and transform them into coupled gateways

* Research partly supported by the EU H2020 RISE programme under the Marie Skłodowska-Curie grant agreement No 778233, by the MIUR project PRIN 2017FTXR7S "IT-MaTTerS" (Methods and Tools for Trustworthy Smart Systems) and by the Project PTR - UNICT 2016-19. The first and second authors have also been partially supported by INdAM as members of GNCS (Gruppo Nazionale per il Calcolo Scientifico). The authors thanks the reviewers for their helpful comments and also M. Dezani for her support.

connecting the two systems. More precisely, if a message is sent to one of the gateways, it is forwarded to the other gateway, which sends it to the other system.

Of course, for such a composition to be well-behaved, the two gateways should exhibit behaviours which are essentially dual of each other: when one wants to send a message the other one needs to be willing to receive the same message. Such an intuition has been formalised as a *compatibility* relation. It has also been shown that compatibility, together with conditions of no mixed states and ?!-determinism on the selected participants, ensures that the composition is well-behaved. For instance, if the components are deadlock-free then the system resulting from the composition is deadlock-free too.

In this paper we first revise such results in a setting of synchronous CFSMs, while [3,4] focus on the asynchronous FIFO case. Somehow surprisingly, stricter conditions are required to ensure compositionality of deadlock freedom. We then propose a new composition methodology which replaces the two selected participants with a unique gateway. Beyond saving some communications and simplifying the analysis, this second methodology is also more general since the conditions needed for compositionality of deadlock freedom are slightly weaker. We call this second composition *semi-direct*, to distinguish it also from direct composition as proposed in [5] in a context of multiparty session types [17], which avoids the need for gateways altogether. Notably, two-gateways composition is completely transparent for the participants different from the interface ones, semi-direct composition requires renaming some of their communications, while direct composition may require a non-trivial restructuring of their behaviours.

Structure of the paper. Section 2 introduces systems of CFSMs and related notions. Composition by gateways and semi-direct composition are discussed in Section 3 and Section 4, respectively. Conclusions, related and future work are discussed in Section 5.

2 Background

Communicating Finite State Machines (CFSMs) [10] are Finite State Automata (FSAs) where transitions are labelled by communications.

Definition 2.1 (FSA). *A* Finite State Automaton *(FSA) is a tuple* $A = \langle \mathcal{S}, s_0, \mathcal{L}, \rightarrow \rangle$ *where*

- $\mathcal{S}$ *is a finite set of states (ranged over by* $s, q, \ldots$*);*
- $s_0 \in \mathcal{S}$ *is the* initial state*;*
- $\mathcal{L}$ *is a finite set of labels (ranged over by* $l, \lambda, \ldots$*);*
- $\rightarrow \subseteq \mathcal{S} \times \mathcal{L} \times \mathcal{S}$ *is a set of transitions.*

We use the usual notation $s_1 \xrightarrow{\lambda} s_2$ *for the transition* $(s_1, \lambda, s_2) \in \rightarrow$*, and* $s_1 \rightarrow s_2$ *when there exists* λ *such that* $s_1 \xrightarrow{\lambda} s_2$*, as well as* $\rightarrow^*$ *for the reflexive and transitive closure of* $\rightarrow$*. The set of* reachable states *of* A *is* $\mathcal{R}(A) = \{\, s \mid s_0 \rightarrow^* s \,\}$.

Let $s \xrightarrow{\lambda} s' \in A$ emphasise that the transition belongs to (the set of transitions of) an FSA A; likewise, $q \in A$ stands for "q belongs to the states of A". A transition $s \xrightarrow{\lambda} s'$ (resp. $s' \xrightarrow{\lambda} s$) is an *outgoing* (resp. *incoming*) transition of s. We write $f[x \mapsto y]$ for the update of the function f in a point x of its domain with the value y. Also, $\mathrm{dom}(f)$ denotes the domain of the function f.

We now define systems of CFSMs, by adapting the definitions in [10] to our context. Let $\mathfrak{P}$ be a set of *participants* (or *roles*, ranged over by A, B, etc.) and $\mathcal{M}$ a set of *messages* (ranged over by m, n, etc.). We take $\mathfrak{P}$ and $\mathcal{M}$ disjoint.

Definition 2.2 (Communicating system). *A* communicating finite-state machine *(CFSM) is an FSA with labels in the set*

$$\mathcal{L}_{act} = \{\mathsf{A}\,\mathsf{B}!\mathsf{m}, \mathsf{A}\,\mathsf{B}?\mathsf{m} \mid \mathsf{A} \neq \mathsf{B} \in \mathfrak{P}, \mathsf{m} \in \mathcal{M}\}$$

of actions. *The* subject *of an output (resp. input) action* $\mathsf{A}\,\mathsf{B}!\mathsf{m}$ *(resp.* $\mathsf{A}\,\mathsf{B}?\mathsf{m}$*) is* A *(resp.* B*). A CFSM is* A-local *if all its transitions have subject* A.

A (communicating) system *is a map* $\mathsf{S} = (M_\mathsf{A})_{\mathsf{A} \in \mathcal{P}}$ *assigning an* A*-local CFSM* M_A *to each participant* $\mathsf{A} \in \mathcal{P}$ *where* $\mathcal{P} \subseteq \mathfrak{P}$ *is finite and any participant occurring in a transition of* M_A *is in* $\mathcal{P}$.

Note that systems satisfying the above definition are *closed*: in fact any input or output action does refer to participants belonging to the system itself.

We now define, following [7,6], the synchronous semantics of systems of CFSMs, which is itself an FSA (differently from the asynchronous case, where the set of states can be infinite).

Definition 2.3 (Synchronous semantics). *Let* S *be a communicating system. A* synchronous configuration of S *is a map* $s = (q_\mathsf{A})_{\mathsf{A} \in \mathrm{dom}(\mathsf{S})}$ *assigning a* local state $q_\mathsf{A} \in \mathsf{S}(\mathsf{A})$ *to each* $\mathsf{A} \in \mathrm{dom}(\mathsf{S})$.

The synchronous semantics *of* S *is the FSA* $[\![\mathsf{S}]\!] = \langle \mathcal{S}, s_0, \mathcal{L}_{int}, \to \rangle$ *where*

- $\mathcal{S}$ *is the set of synchronous configurations of* S*, as defined above;*
- $s_0 = (q_{0\mathsf{A}})_{\mathsf{A} \in \mathrm{dom}(\mathsf{S})} \in \mathcal{S}$ *is the* initial *configuration where, for each* $\mathsf{A} \in \mathrm{dom}(\mathsf{S})$, $q_{0\mathsf{A}}$ *is the initial state of* $\mathsf{S}(\mathsf{A})$*;*
- $\mathcal{L}_{int} = \{\mathsf{A} \to \mathsf{B} \colon \mathsf{m} \mid \mathsf{A} \neq \mathsf{B} \in \mathfrak{P}$ *and* $\mathsf{m} \in \mathcal{M}\}$ *is a set of interaction labels;*
- $s \xrightarrow{\mathsf{A} \to \mathsf{B} \colon \mathsf{m}} s[\mathsf{A} \mapsto q, \mathsf{B} \mapsto q'] \in [\![\mathsf{S}]\!]$ *if* $s(\mathsf{A}) \xrightarrow{\mathsf{A}\,\mathsf{B}!\mathsf{m}} q \in \mathsf{S}(\mathsf{A})$ *and* $s(\mathsf{B}) \xrightarrow{\mathsf{A}\,\mathsf{B}?\mathsf{m}} q' \in \mathsf{S}(\mathsf{B})$.

We say that s enables $q \xrightarrow{\mathsf{A}\,\mathsf{B}!\mathsf{m}} q' \in \mathsf{S}(\mathsf{A})$ *(resp.* $q \xrightarrow{\mathsf{B}\,\mathsf{A}?\mathsf{m}} q' \in \mathsf{S}(\mathsf{A})$*) when* $s(\mathsf{A}) = q$.

As expected, an interaction $\mathsf{A} \to \mathsf{B} \colon \mathsf{m}$ occurs when A performs an output $\mathsf{A}\,\mathsf{B}!\mathsf{m}$ and B the corresponding input $\mathsf{A}\,\mathsf{B}?\mathsf{m}$.

As discussed in the Introduction, in this paper we will study preservation of communication properties under composition. As sample property we choose the well-known notion of deadlock freedom. The definition below adapts the one in [13] to a synchronous setting (as done also in [20,27]).

Definition 2.4 (Deadlock freedom). *Let* S *be a communicating system. A configuration* $s \in \mathcal{R}([\![\mathsf{S}]\!])$ *is a* deadlock *if*

– *s has no outgoing transitions in $[\![\mathsf{S}]\!]$ and*
– *there exists $\mathsf{A} \in \mathcal{P}$ such that $s(\mathsf{A})$ has an outgoing transition in $\mathsf{S}(\mathsf{A})$.*

System S is deadlock-free *if for each $s \in \mathcal{R}([\![\mathsf{S}]\!])$, s is not a deadlock.*

3 Composition via Gateways

This section discusses composition of systems of CFSMs via gateways, as introduced in [3,4], and studies its properties under the synchronous semantics. The main idea is that two systems of CFSMs, say S_1 and S_2, can be composed by transforming one participant in each of them into gateways connected to each other. Let us call H the selected participant in S_1 and K the one in S_2. The gateways for H and K are connected to each other and act as forwarders: each message sent to the gateway for H by a participant from the original system S_1 is now forwarded to the gateway for K, that in turn forwards it to the same participant to which K sent it in the original system S_2. The dual will happen to messages that the gateway for K receives from S_2. A main advantage of this approach is that no extension of the CFSM model is needed to transform systems of CFSMs, which are normally closed systems, into open systems that can be composed. Another advantage is that the composition is fully transparent to all participants different from H and K.

We will now define composition via gateways on systems of CFSMs, following the intuition above.

Definition 3.1 (Gateway). *Given a H-local CFSM M and a participant K, the* gateway of M towards K *is the CFSM* $\mathsf{gw}(M, \mathsf{K})$ *obtained by replacing:*

– *each transition $q \xrightarrow{\mathsf{H}\,\mathsf{A}!\mathsf{m}} q' \in M$ with $q \xrightarrow{\mathsf{K}\,\mathsf{H}?\mathsf{m}} q'' \xrightarrow{\mathsf{H}\,\mathsf{A}!\mathsf{m}} q'$ for some fresh state q'';*
– *each transition $q \xrightarrow{\mathsf{A}\,\mathsf{H}?\mathsf{m}} q' \in M$ with $q \xrightarrow{\mathsf{A}\,\mathsf{H}?\mathsf{m}} q'' \xrightarrow{\mathsf{H}\,\mathsf{K}!\mathsf{m}} q'$ for some fresh state q''.*

We compose systems with disjoint participants through two of them, say H and K, by taking all the participants of the original systems but H and K, whereas H and K are replaced by their respective gateways.

Definition 3.2 (System composition). *Let S_1 and S_2 be two systems with disjoint domains. The* composition of S_1 and S_2 via $\mathsf{H} \in \mathrm{dom}(\mathsf{S}_1)$ and $\mathsf{K} \in \mathrm{dom}(\mathsf{S}_2)$ *is defined as*

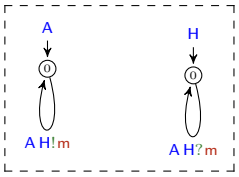
$$\mathsf{S}_1 \,{}^{\mathsf{H}\leftrightarrow\mathsf{K}}\, \mathsf{S}_2 = \mathsf{A} \mapsto \begin{cases} \mathsf{S}_i(\mathsf{A}), & \text{if } \mathsf{A} \in \mathrm{dom}(\mathsf{S}_i) \setminus \{\mathsf{H}, \mathsf{K}\} \text{ with } i \in \{1,2\} \\ \mathsf{gw}(\mathsf{S}_1(\mathsf{H}), \mathsf{K}), & \text{if } \mathsf{A} = \mathsf{H} \\ \mathsf{gw}(\mathsf{S}_2(\mathsf{K}), \mathsf{H}), & \text{if } \mathsf{A} = \mathsf{K} \end{cases}$$

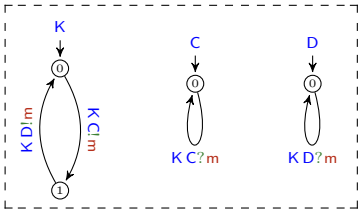
(Note that $\mathrm{dom}(\mathsf{S}_1 \,{}^{\mathsf{H}\leftrightarrow\mathsf{K}}\, \mathsf{S}_2) = \mathrm{dom}(\mathsf{S}_1) \cup \mathrm{dom}(\mathsf{S}_2)$*.)*

We remark again that, by the above approach for composition, we do not actually need to formalise the notion of *open* system. In fact any closed system can be looked at as open by choosing (according to the current necessities) two suitable participants in the "to-be-connected" systems and transforming them into two forwarders.

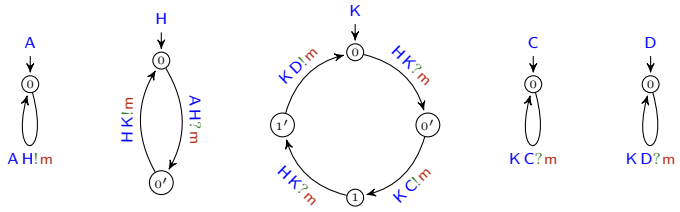
We also note that the notion of composition above is structural: a corresponding notion of behavioural composition has been studied in [5] in a context of multiparty session types [17].

Example 3.3. Take the systems S_1 and S_2 below

$\mathsf{S}_1 =$ 

$\mathsf{S}_2 =$ 

The system consisting of the following CFSMs



is the composition $\mathsf{S}_1 {}^{\mathsf{H}\leftarrow\!\!\!\!\times\mathsf{K}} \mathsf{S}_2$. ◇

Given a configuration of the composition of systems S_1 and S_2 we can retrieve the configurations of the two subsystems by taking only the states of participants in S_i (for $i \in \{1, 2\}$) while avoiding, for the gateways, to take the fresh states introduced by the gateway construction.

Definition 3.4 (Configuration projection). *Let s be a configuration of a composed system $\mathsf{S}_1 {}^{\mathsf{H}\leftarrow\!\!\!\!\times\mathsf{K}} \mathsf{S}_2$. The* projection of s on S_1 *is the map $s|_1$ defined by*

$$s|_1 : \mathsf{A} \mapsto \begin{cases} s(\mathsf{A}), & \text{if } s(\mathsf{A}) \text{ is not fresh} \\ q, & \text{if } \mathsf{A} = \mathsf{H},\ s(\mathsf{H}) \text{ is fresh and } q \xrightarrow{\mathsf{K}\,\mathsf{H}?\mathsf{m}} s(\mathsf{H}) \in M \\ q, & \text{if } \mathsf{A} = \mathsf{H},\ s(\mathsf{H}) \text{ is fresh and } s(\mathsf{H}) \xrightarrow{\mathsf{H}\,\mathsf{K}!\mathsf{m}} q \in M \end{cases}$$

where $M = \mathsf{gw}(\mathsf{S}_1(\mathsf{H}), \mathsf{K})$. The definition for $s|_2$ is analogous.

Intuitively, in the projection $s|_1$, if H is in a fresh state after receiving from K, then the other participants in S_1 are still not aware of the message arrival, hence to have a coherent configuration we take the state of H before the receive. If instead H is in a fresh state before sending to K, then the other participants in S_1 know that the message has been sent, hence to have a coherent configuration we take the state of H after the send. (A similar intuition applies to $s|_2$.)

Example 3.5. Let us consider the system $\mathsf{S} = \mathsf{S}_1 {}^{\mathsf{H}\leftrightarrow\mathsf{K}} \mathsf{S}_2$ of Example 3.3. Take its configuration $s = (0_\mathsf{A}, 0_\mathsf{H}, 1'_\mathsf{K}, 0_\mathsf{C}, 0_\mathsf{D})$. It is easy to check that $s \in \mathcal{R}(\llbracket \mathsf{S} \rrbracket)$. In fact
$s_0 = (0_\mathsf{A}, 0_\mathsf{H}, 0_\mathsf{K}, 0_\mathsf{C}, 0_\mathsf{D}) \xrightarrow{\mathsf{A}\to\mathsf{H}:\,\mathsf{m}} (0_\mathsf{A}, 0'_\mathsf{H}, 0_\mathsf{K}, 0_\mathsf{C}, 0_\mathsf{D}) \xrightarrow{\mathsf{H}\to\mathsf{K}:\,\mathsf{m}} (0_\mathsf{A}, 0_\mathsf{H}, 0'_\mathsf{K}, 0_\mathsf{C}, 0_\mathsf{D})$
$\xrightarrow{\mathsf{K}\to\mathsf{C}:\,\mathsf{m}} (0_\mathsf{A}, 0_\mathsf{H}, 1_\mathsf{K}, 0_\mathsf{C}, 0_\mathsf{D}) \xrightarrow{\mathsf{A}\to\mathsf{H}:\,\mathsf{m}} (0_\mathsf{A}, 0'_\mathsf{H}, 1_\mathsf{K}, 0_\mathsf{C}, 0_\mathsf{D}) \xrightarrow{\mathsf{H}\to\mathsf{K}:\,\mathsf{m}} (0_\mathsf{A}, 0_\mathsf{H}, 1'_\mathsf{K}, 0_\mathsf{C}, 0_\mathsf{D})$
The projections of s on, respectively, S_1 and S_2 are

$$s|_1 = (0_\mathsf{A}, 0_\mathsf{H}) \quad \textit{and} \quad s|_2 = (1_\mathsf{K}, 0_\mathsf{C}, 0_\mathsf{D})$$

Notice that (as we shall prove in Proposition 3.11), from $s \in \mathcal{R}(\llbracket \mathsf{S} \rrbracket)$ it is possible to infer that $s|_1 \in \mathcal{R}(\llbracket \mathsf{S}_1 \rrbracket)$ and $s|_2 \in \mathcal{R}(\llbracket \mathsf{S}_2 \rrbracket)$. ◇

Being able to build the composition via gateways does not ensure that the result is well-behaved or that its behaviour is related in any way to the behaviour of the original systems. We provide below sufficient conditions for this to happen. We focus in particular on whether deadlock freedom is preserved under composition. Somehow surprisingly, in the synchronous case preservation of deadlock freedom requires stricter conditions than in the asynchronous one.

Informally, two CFSMs M_1 and M_2 are *compatible* if M_1 is bisimilar to the dual of M_2 provided that the communicating partners are abstracted away. In order to define compatibility, a few simple definitions are handy.
Let $\mathcal{L}_{\mathrm{i/o}} = \{\ ?\mathsf{m}, !\mathsf{m} \mid \mathsf{m} \in \mathcal{M}\ \}$ and define the functions

$$\mathsf{io} : \mathcal{L}_{\mathrm{act}} \to \mathcal{L}_{\mathrm{i/o}} \qquad \text{and} \qquad \overline{(\cdot)} : \mathcal{L}_{\mathrm{i/o}} \to \mathcal{L}_{\mathrm{i/o}}$$

by the following clauses

$$\mathsf{io}(\mathsf{A}\,\mathsf{B}?\mathsf{m}) = ?\mathsf{m} \qquad \mathsf{io}(\mathsf{A}\,\mathsf{B}!\mathsf{m}) = !\mathsf{m} \qquad \text{and} \qquad \overline{?\mathsf{m}} = !\mathsf{m} \qquad \overline{!\mathsf{m}} = ?\mathsf{m}$$

which extend to CFSMs in the obvious way: given a CFSM $M = \langle \mathcal{S}, q_0, \mathcal{L}_{\mathrm{act}}, \to \rangle$, we define $\mathsf{io}(M) = \langle \mathcal{S}, q_0, \mathcal{L}_{\mathrm{i/o}}, \to' \rangle$ where $\to' = \{\ q \xrightarrow{\mathsf{io}(l)} q' \mid q \xrightarrow{l} q' \in M\ \}$; and likewise for $\overline{M}$.

Definition 3.6 (Compatibility). *Two CFSMs M_1 and M_2 are* compatible *if $\mathsf{io}(M_1)$ is bisimilar to $\overline{\mathsf{io}(M_2)}$. Given two communicating systems S_1 and S_2, participants $\mathsf{H} \in \mathrm{dom}(\mathsf{S}_1)$ and $\mathsf{K} \in \mathrm{dom}(\mathsf{S}_2)$ are* compatible roles *if $\mathsf{S}_1(\mathsf{H})$ and $\mathsf{S}_2(\mathsf{K})$ are compatible CFSMs.*

We refer to the bisimilarity in Definition 3.6 as *compatibility bisimilarity*. Note that the compatibility bisimilarity between M_1 and M_2 is a relation between their states. It is easy to check that H and K of Example 3.3 are compatible roles.

Definition 3.7. *An A-local CFSM M is:*

i) ?-deterministic *(resp.* !-deterministic*) if $q \xrightarrow{\mathsf{X}\,\mathsf{A}?\mathsf{m}} q'$ and $q \xrightarrow{\mathsf{Y}\,\mathsf{A}?\mathsf{m}} q'' \in M$ (resp. $q \xrightarrow{\mathsf{A}\,\mathsf{X}!\mathsf{m}} q'$ and $q \xrightarrow{\mathsf{A}\,\mathsf{Y}!\mathsf{m}} q'' \in M$) implies $q' = q''$;*
ii) ?!-deterministic *if it is both ?-deterministic and !-deterministic;*
iii) mixed-deterministic *if $\mathsf{m} \neq \mathsf{n}$ for all $q \xrightarrow{\mathsf{X}\,\mathsf{A}?\mathsf{m}} q'$ and $q \xrightarrow{\mathsf{A}\,\mathsf{Y}!\mathsf{n}} q'' \in M$.*

A state $q \in M$ is a *sending* (resp. *receiving*) state if it has outgoing transitions, all of which are labelled with sending (resp. receiving) actions; q is a *mixed* state if it has outgoing transitions and q is neither sending nor receiving.

Definition 3.8 ((H,K)-composability). *Two systems S_1 and S_2 with disjoint domains are (H,K)-*composable *if $\mathsf{H} \in \mathrm{dom}(\mathsf{S}_1)$ and $\mathsf{K} \in \mathrm{dom}(\mathsf{S}_2)$ are two compatible roles whose machines have no mixed states and are ?!-deterministic.*

Definition 3.9. *Let $\mathsf{gw}(M_\mathsf{H},\mathsf{K})$ be a gateway extracted from an H-local CFSM. Function $\mathsf{nof}_{M_\mathsf{H}}(\cdot)$ maps the states of $\mathsf{gw}(M_\mathsf{H},\mathsf{K})$ to the states of M_H as follows:*

$$\mathsf{nof}_{M_\mathsf{H}}(q) = \begin{cases} q & \text{if } q \text{ is not fresh} \\ q' & \text{if } q \text{ is fresh and } q' \xrightarrow{\mathsf{A}\,\mathsf{H}?\mathsf{m}} q \in \mathsf{gw}(M_\mathsf{H},\mathsf{K}) \text{ for some } \mathsf{A},\mathsf{m} \\ q' & \text{if } q \text{ is fresh and } q \xrightarrow{\mathsf{H}\,\mathsf{A}!\mathsf{m}} q' \in \mathsf{gw}(M_\mathsf{H},\mathsf{K}) \text{ for some } \mathsf{A},\mathsf{m} \end{cases}$$

Lemma 3.10. *Function $\mathsf{nof}_{M_\mathsf{H}}$ is well-defined.*

Proof. The restriction of $\mathsf{nof}_{M_\mathsf{H}}$ to the states of M_H is the identity. If q is not a state of M_H, then it is fresh by definition of $\mathsf{gw}(M_\mathsf{H},\mathsf{K})$. By definition of $\mathsf{gw}(M_\mathsf{H},\mathsf{K})$ again, there is a unique q' such that either $q' \xrightarrow{\mathsf{A}\,\mathsf{H}?\mathsf{m}} q \in \mathsf{gw}(M_\mathsf{H},\mathsf{K})$ or $q \xrightarrow{\mathsf{H}\,\mathsf{A}!\mathsf{m}} q' \in \mathsf{gw}(M_\mathsf{H},\mathsf{K})$. □

In the system $\mathsf{S} = \mathsf{S}_1 \overset{\mathsf{H}\leftrightarrow\mathsf{K}}{} \mathsf{S}_2$ of Example 3.3 it is easy to check, for example, that $\mathsf{nof}_{\mathsf{S}(\mathsf{H})}(0) = 0$ and $\mathsf{nof}_{\mathsf{S}(\mathsf{K})}(1') = 0$.

Function $\mathsf{nof}_{M_\mathsf{H}}$ is close to the definition of configuration projection (but for considering a single state instead of a whole configuration) with a main change. Indeed, when $\mathsf{gw}(M_\mathsf{H},\mathsf{K})$ receives a message from its own system S_1 going to some fresh state q'', configuration projection maps it to the next state, since the rest of S_1 is aware of the transition but $\mathsf{gw}(M_\mathsf{H},\mathsf{K})$ will complete the transition only in the next state. Instead, function $\mathsf{nof}_{M_\mathsf{H}}$ maps q'' to the previous state since S_2, and K in particular, are not yet aware of the transition. Thus, function $\mathsf{nof}_{M_\mathsf{H}}$ is designed to establish a correspondence with the other system as shown by the next proposition.

Proposition 3.11. *Let $\mathsf{S} = \mathsf{S}_1 \overset{\mathsf{H}\leftrightarrow\mathsf{K}}{} \mathsf{S}_2$ be the composition of two (H,K)-composable systems S_1 and S_2. If $s \in \mathcal{R}(\llbracket \mathsf{S} \rrbracket)$ then exactly one of the following cases hold for $q_\mathsf{H} = s(\mathsf{H})$ and $q_\mathsf{K} = s(\mathsf{K})$, the states in s of the gateway CFSMs:*

1. *both q_H and q_K are not fresh;*
2. *either q_H is fresh, q_K is not fresh, $q_\mathsf{H} \xrightarrow{\mathsf{H}\,\mathsf{K}!\mathsf{m}} q \in \mathsf{S}_1(\mathsf{H})$, or, symmetrically, q_K is fresh, q_H is not fresh, $q_\mathsf{K} \xrightarrow{\mathsf{K}\,\mathsf{H}!\mathsf{m}} q \in \mathsf{S}_2(\mathsf{K})$;*
3. *either q_H is fresh, q_K is not fresh, and there is $\mathsf{A} \in \mathrm{dom}(\mathsf{S}_1)$ such that $q_\mathsf{H} \xrightarrow{\mathsf{H}\,\mathsf{A}!\mathsf{m}} q \in \mathsf{S}_1(\mathsf{H})$, or, symmetrically, q_K is fresh, q_H is not fresh, and there is $\mathsf{B} \in \mathrm{dom}(\mathsf{S}_2)$ such that $q_\mathsf{K} \xrightarrow{\mathsf{K}\,\mathsf{B}!\mathsf{m}} q \in \mathsf{S}_2(\mathsf{K})$;*
4. *both q_H and q_K are fresh and either $q_\mathsf{H} \xrightarrow{\mathsf{H}\,\mathsf{K}!\mathsf{m}} q \in \mathsf{S}_1(\mathsf{H})$, and there is $\mathsf{A} \in \mathrm{dom}(\mathsf{S}_2)$ such that $q_\mathsf{K} \xrightarrow{\mathsf{K}\,\mathsf{B}!\mathsf{n}} q \in \mathsf{S}_2(\mathsf{K})$, or, symmetrically, $q_\mathsf{K} \xrightarrow{\mathsf{K}\,\mathsf{H}!\mathsf{m}} q \in \mathsf{S}_2(\mathsf{K})$, and there is $\mathsf{A} \in \mathrm{dom}(\mathsf{S}_1)$ such that $q_\mathsf{H} \xrightarrow{\mathsf{H}\,\mathsf{A}!\mathsf{n}} q \in \mathsf{S}_1(\mathsf{H})$.*

Also, $s|_1$ is reachable in S_1, $s|_2$ is reachable in S_2 and $\mathsf{nof}_{M_\mathsf{H}}(q_\mathsf{H}) \sim \mathsf{nof}_{M_\mathsf{K}}(q_\mathsf{K})$.

Proof. The proof is by induction on the number n of transitions performed to reach s. If $n = 0$ then by construction we are in case 1. The conditions on configurations and on bisimulation hold by construction.

Let us assume that we are in one of the cases above and a further transition is performed. Since composition is symmetric for each possibility we do not detail the symmetric case. Also note that in each of the cases, if the transition does not involve the gateways, then we are still in the same case. The condition on configurations hold since the same step can be taken by the same participants in one of the two systems, and the ones of the other system do not move. The condition on bisimulation holds since the state of the gateways does not change.

If we were in case 1, a transition involving a gateway has necessarily the form $s \xrightarrow{\mathsf{A}\longrightarrow\mathsf{H}:\,\mathsf{m}} s'$ (or a similar one for the gateway for K) since an output from a gateway would require a gateway to be in a fresh state. This leads us to case 2. Indeed, the gateway for H goes to the fresh state $s'(\mathsf{H})$ and is willing to execute the gateway communication $s'(\mathsf{H}) \xrightarrow{\mathsf{H}\,\mathsf{K}!\mathsf{m}} q$, while the state of the gateway for K does not change. The condition on configurations hold by induction on $s|_2$ and holds on $s|_1$ since $s'(\mathsf{H})|_1 = q$ and by construction $s|_1 \xrightarrow{\mathsf{A}\longrightarrow\mathsf{H}:\,\mathsf{m}} s'|_1$. The condition on bisimulation holds by inductive hypothesis since $\mathsf{nof}_{M_\mathsf{H}}(q) = s|_1(\mathsf{H})$ and the state of the gateway for K does not change.

In case 2 a transition involving the gateway necessarily is the gateway communication $s \xrightarrow{\mathsf{H}\longrightarrow\mathsf{K}:\,\mathsf{m}} s'$, leading us to case 3. Indeed, the gateway for H cannot perform other transitions and thanks to the condition on compatibility and the fact that gateway roles do not have mixed states K cannot perform any input from its system. Thus, the gateway for H goes to a non-fresh state while the gateway for K goes to a fresh state, willing to execute an output $s'(\mathsf{K}) \xrightarrow{\mathsf{K}\,\mathsf{B}!\mathsf{m}} q$ towards its system. The condition on configurations holds by inductive hypothesis since projection generates the same configurations as before. The condition on bisimulation holds since the two participants take corresponding steps. The resulting states are in a correspondence thanks to ?!-determinism.

In case 3 for a transition involving a gateway there are two possibilities, according to whether the gateway taking a transition is in a fresh state or not.

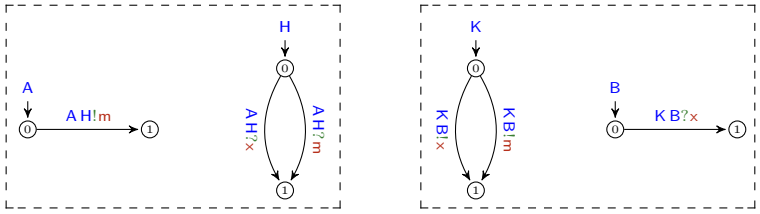
– The gateway in a fresh state, say K, takes a transition. By construction it delivers a message m to a participant in its system via a transition $s \xrightarrow{\mathsf{K}\longrightarrow\mathsf{B}:\,\mathsf{m}} s'$. This leads us to case 1. Indeed, K goes to a non-fresh state, while H was in a non-fresh state by hypothesis and does not move. The condition on configurations holds by inductive hypothesis for $s|_1$ and holds for $s|_2$ since B can do the same move as in s and the two moves of the gateway for K (the gateway transition which was not taken into account yet and the delivery of message m) correspond to the complementary move of K in $s|_2$. The condition on bisimulation holds by inductive hypothesis since $\mathsf{nof}_{M_\mathsf{H}}(\cdot)$ projects on the same states as before the transition.

– The gateway in a non-fresh state, say H, takes a transition. By construction it takes a message from its own system via a transition $s \xrightarrow{\mathsf{A}\to\mathsf{H}:\,\mathsf{n}} s'$. This leads us to case 4. Indeed, H goes to a fresh state while K was already in a fresh state. Then $s'(\mathsf{K}) \xrightarrow{\mathsf{K}\,\mathsf{B}!\mathsf{m}} q$ by inductive hypothesis and $s'(\mathsf{H}) \xrightarrow{\mathsf{H}\,\mathsf{K}!\mathsf{n}} q'$ by construction. The reasoning on conditions of configurations and bisimulation is similar to the one of a message taken by the gateway in case 1.

In case 4, when a transition involving a gateway, say K, is performed, it is necessarily of the form $s \xrightarrow{\mathsf{K}\to\mathsf{B}:\,\mathsf{m}} s'$ since the gateway for K by construction cannot take any other action. This leads to case 2. Indeed, H remains in a fresh state and willing to execute a transition $s'(\mathsf{H}) \xrightarrow{\mathsf{H}\,\mathsf{K}!\mathsf{m}} q$ while K goes to a non-fresh state. The reasoning for the conditions on configurations and bisimulation is similar to the one for case 3 when K delivers a message to its system. □

Now, one may think that analogously to what happens in [3,4], if two systems are (H,K)-composable and deadlock-free then their composition is deadlock-free too. Unfortunately, this is not the case, as shown by the examples below. The first example is based on an example in [4], that shows that mixed states have to be forbidden and that holds for the synchronous case as well. In the synchronous case, however, we can also exchange some inputs with outputs and obtain the same behaviour *without* mixed states.

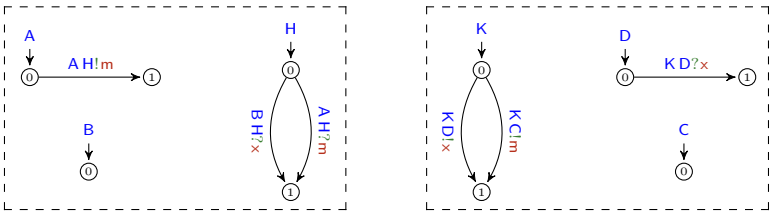
Example 3.12. Take the following CFSMs



and consider the composition of the system with participants A and H with the one with participants K and B. Clearly, the two systems are (H,K)-composable and deadlock-free, yet their composition has a deadlock; in fact, when the gateway for K receives m, participant B is waiting only for x. By considering the second system alone, this is not a deadlock, since B forces K to select the right branch.

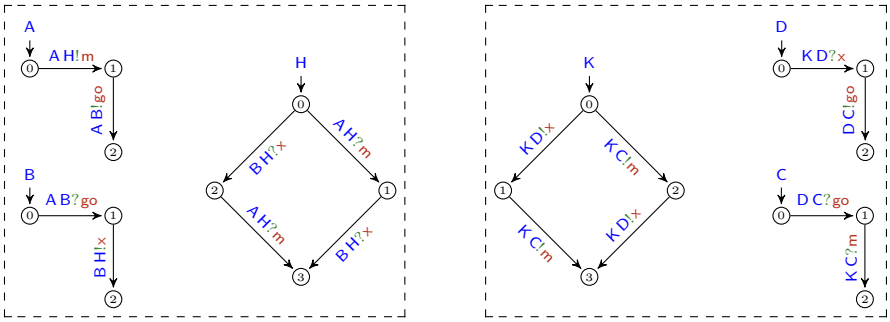
Note that the situation would be different in an asynchronous setting. Indeed, the second system could deadlock. This is due to the fact that K could send m without synchronising with B. ◇

Example 3.13. Take the CFSMs below



The same reasoning of Example 3.12 can be applied here, to systems with participants $\mathsf{A}, \mathsf{B}, \mathsf{H}$ and $\mathsf{C}, \mathsf{D}, \mathsf{K}$. Hence, choices made by different participants are problematic as well. ◇

Example 3.14. Take the CFSMs below



The reasoning is again similar, and shows that the composition of systems $\mathsf{A}, \mathsf{B}, \mathsf{H}$ and $\mathsf{K}, \mathsf{C}, \mathsf{D}$ deadlocks while the two systems in isolation do not. Hence also concurrency diamonds are problematic. ◇

Given the examples above, it is clear that, differently from the asynchronous case, deadlock freedom can be preserved only under very strict conditions on interface participants. Indeed we show below that it can be preserved if interface participants do not contain choices.

Definition 3.15 (Sequential CFSM). *A CFSM is* sequential *if each of its states has at most one outgoing transition.*

It is immediate to check that $\mathsf{gw}(M, \mathsf{H})$ is sequential if M is so. Moreover, trivially, a sequential M is also ?!-deterministic and with no mixed state (and hence mixed-deterministic).

Theorem 3.16 (Deadlock freedom for sequential interfaces). *Let* S_1 *and* S_2 *be two* (H, K)*-composable and deadlock-free systems, such that* $\mathsf{S}_1(\mathsf{H})$ *and* $\mathsf{S}_2(\mathsf{K})$ *are sequential. Then the composed system* $\mathsf{S}_1 \,{}^{\mathsf{H}\leftrightarrow\mathsf{K}}\, \mathsf{S}_2$ *is deadlock-free.*

Proof. We show that if the composed system $\mathsf{S}_1 \,{}^{\mathsf{H}\leftrightarrow\mathsf{K}}\, \mathsf{S}_2$ reaches a deadlock configuration s then at least one of $s|_1$ and $s|_2$ is a deadlock. First, we show that if a participant A (say from S_1) is willing to take an action in a configuration s of the composed system then some participant is willing to take an action in $s|_1$ or in $s|_2$ (and the same for participants in S_2). Note that $s|_1$ and $s|_2$ are reachable in, respectively, S_1 and S_2 thanks to the condition on configurations in Proposition 3.11.

If $\mathsf{A} \neq \mathsf{H}$ then A wants to take the same action by definition of $s|_1$.

If $\mathsf{A} = \mathsf{H}$ and it is willing to receive from K then, by the definition of $s|_1$ and of gateway, H in $s|_1$ is willing to send a message to some participant in S_1.

If $\mathsf{A} = \mathsf{H}$ and it is willing to send to K then $\mathsf{nof}_{M_\mathsf{H}}(s(\mathsf{H})) \sim \mathsf{nof}_{M_\mathsf{K}}(s(\mathsf{K}))$ by the condition on bisimulation of Proposition 3.11. By definition of $\mathsf{nof}_{M_\mathsf{H}}(\cdot)$ and

of gateway, $\mathsf{nof}_{M_\mathsf{H}}(s(\mathsf{H}))$ is willing to take a message from its own system, hence $\mathsf{nof}_{M_\mathsf{K}}(s(\mathsf{K}))$ is willing to send such a message to its own system thanks to the definition of bisimulation. By definition of configuration projection $\mathsf{nof}_{M_\mathsf{K}}(s(\mathsf{K}))$ is also a state in $s|_2$, hence there is a participant willing to take a transition.

Now we show that if no transition is enabled in a configuration s of the composed system then no transition is enabled in $s|_1$ and $s|_2$. We prove the contrapositive, showing that if there is an enabled transition in $s|_1$ or in $s|_2$ then there is a transition enabled in s as well. There are a few cases to consider.

Transition not involving the interface roles: this case follows immediately from the definition of system transition and of configuration projection.

Communication towards an interface role: let A be the sender and H the interface role. The transition is of the form $s|_1 \xrightarrow{\mathsf{A}\longrightarrow\mathsf{H}:\,\mathsf{m}} s_1$. There are two possibilities. If the gateway for H is not in a fresh state in s then the same transition can trigger in the composed system thanks to the definition of system transition and of configuration projection.

If it is in a fresh state then thanks to the definition of gateway and of configuration projection it still needs to complete a previous gateway communication. The other gateway, K, may be in a fresh state or not. If it is not, thanks to the definition of $\mathsf{nof}_{M_\mathsf{H}}(\cdot)$ and to the condition on bisimulation of Proposition 3.11 it is willing to accept the gateway communication which can thus trigger as desired. If K is in a fresh state then thanks to the definition of configuration projection and of gateway it is willing to deliver a message to S_2. Since S_2 is not a deadlock and a participant is willing to take a transition then a transition can trigger in S_2 too. Thus, we can apply the other cases to find a witness transition in the composed system. Note that the transition in S_2 cannot be towards an interface role thanks to the condition on bisimulation of Proposition 3.11 and since there are no mixed states, hence this reasoning does not cycle.

Communication from an interface role: let K be the interface role and B the target participant. Hence, the transition is of the form $s|_2 \xrightarrow{\mathsf{K}\longrightarrow\mathsf{B}:\,\mathsf{m}} s_2$. Thanks to the definition of gateway and of system projection the gateway for K in s is either willing to deliver a message to some participant in S_2 or to receive from the gateway for H. In the first case, since the gateway is sequential then the participant is B and the message m, hence the transition can trigger.

If K is willing to receive from H then thanks to the definition of $\mathsf{nof}_{M_\mathsf{H}}(\cdot)$ and the condition on bisimulation of Proposition 3.11 then the gateway for H is willing to receive a message from its system or has just received it and is willing to send it through the gateway. In the last case the gateway communication can occur.

If H is willing to receive a message from some participant in S_1 since S_1 is not a deadlock then there is an enabled transition in S_1 as well. Thus, we can apply the other cases to find a witness transition in the composed system. Note that the transition in S_1 cannot be from an interface role thanks to the

condition on bisimulation of Proposition 3.11 and since there are no mixed states, hence this reasoning does not cycle.

Thus, if there is a deadlock configuration s in the composed system then either $s|_1$ or $s|_2$ are deadlocks against the hypothesis. The thesis follows. □

We can infer deadlock-freedom of the system $\mathsf{S} = \mathsf{S}_1 {}^{\mathsf{H}\leftrightarrow\mathsf{K}} \mathsf{S}_2$ of Example 3.3 by the result above, since S_1 and S_1 are (H,K)-composable and deadlock-free, and $\mathsf{S}_1(\mathsf{H})$ and $\mathsf{S}_2(\mathsf{K})$ are sequential.

The result above, however, is not fully satisfying since the sequentiality condition is very strict, but, as shown by Examples 3.12, 3.13, and 3.14, any form of choice is problematic.

However, we can complement the result above with an additional one pinpointing where deadlocks can happen when gateways with choices are allowed: deadlocks can only occur in communications from the gateway to its own system.

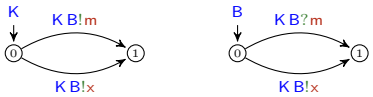
Equivalently, we can drop the sequentiality condition if the systems are such that, whenever their interface role is willing to send a message, the system is ready to receive it. We formalise this condition by the notion of !live participant.

Definition 3.17 (!live participant). *Let* S *be a system and let* $\mathsf{A} \in \mathrm{dom}(\mathsf{S})$. *We say that* $\mathsf{S}(\mathsf{A})$ *is* !live in S *if, for any* $s \in \mathcal{R}([\![\mathsf{S}]\!])$,

$$s(\mathsf{A}) \xrightarrow{\mathsf{A}\,\mathsf{B}!\mathsf{m}} \text{ implies } s \to^* s' \xrightarrow{\mathsf{A}\to\mathsf{B}:\,\mathsf{m}} \text{ for some } s'$$

We remark that !liveness is not a property of the gateway but a property of the system to which it belongs.

It is immediate to check that K is not !live in system S_2 of Example 3.12, whereas K is !live in the following system.



Theorem 3.18 (Deadlock freedom for !live interfaces). *Let* S_1 *and* S_2 *be* (H,K)*-composable and deadlock-free systems. If* $\mathsf{S}_1(\mathsf{H})$ *and* $\mathsf{S}_2(\mathsf{K})$ *are* !*live in, respectively,* S_1 *and* S_2 *then the composed system* $\mathsf{S}_1 {}^{\mathsf{H}\leftrightarrow\mathsf{K}} \mathsf{S}_2$ *is deadlock-free.*

Proof. The proof has the same structure of the one for Theorem 3.16. The only difference is when showing that if there is an enabled transition in $s|_1$ or in $s|_2$ then there is a transition enabled in s as well. Just the case of communication from an interface node changes, in particular when the gateway is willing to deliver some message to some participant in its system. There, !liveness can be used instead of sequentiality to show that indeed some transition can happen. Hence, the thesis follows. □

4 Semi-direct Composition

One may notice that in the form of composition discussed in the previous section the two gateways simply forward messages, and wonder whether they are strictly needed. Indeed, a form of *direct* composition, where gateways are completely avoided, has been studied in [5] in a multiparty session type [17] setting. It has also been shown that applying this technique has a non trivial impact on the participants in the connected systems. We discuss here a different form of composition where a unique gateway is used, and we call it *semi-direct* composition. This has the advantage of saving one gateway and some communications, and also of simplifying some proofs. Moreover, the conditions for deadlock preservation are weaker when non-sequential interfaces are considered (see Theorem 4.10). On the other end, participants in the composed systems are affected, but just by a renaming.

Definition 4.1 (Semi-direct gateway).
Let M_1 and M_2 be, respectively, an H- and a K-local CFSM such that

- *M_1 and M_2 are compatible*
- *for all $q_1 \xrightarrow{l_1} q_1' \in M_1$ and $q_2 \xrightarrow{l_2} q_2' \in M_2$ the participants occurring in l_1 are disjoint from those occurring in l_2*

If W is a fresh role then the semi-direct gateway $\mathsf{sgw}(M_1, \mathsf{W}, M_2)$ *is the CFSM $\langle \mathcal{S}, q_0, \mathcal{L}, \rightarrow \rangle$ such that*

- *$q_0 = (q_1^0, q_2^0)$ with q_1^0 initial state of M_1 and q_2^0 initial state of M_2;*
- *$\mathcal{S}$ includes all pairs (q_1, q_2) and all triples (q_1, m, q_2) such that q_1 is a state of M_1, q_2 of M_2 and m a message which are reachable from the initial state (q_1^0, q_2^0) via the transitions in $\rightarrow$;*
- *$\rightarrow$ includes, for each $q_1 \in M_1$ and $q_2 \in M_2$ related by the compatibility bisimilarity:*
  - *$(q_1, q_2) \xrightarrow{\mathsf{A}\,\mathsf{W}?\mathsf{m}} (q_1', \mathsf{m}, q_2) \xrightarrow{\mathsf{W}\,\mathsf{B}!\mathsf{m}} (q_1', q_2')$ where $q_1 \xrightarrow{\mathsf{A}\,\mathsf{H}?\mathsf{m}} q_1' \in M_1$, $q_2 \xrightarrow{\mathsf{K}\,\mathsf{B}!\mathsf{m}} q_2' \in M_2$,*
  - *$(q_1, q_2) \xrightarrow{\mathsf{B}\,\mathsf{W}?\mathsf{m}} (q_1, \mathsf{m}, q_2') \xrightarrow{\mathsf{W}\,\mathsf{A}!\mathsf{m}} (q_1', q_2')$ where $q_2 \xrightarrow{\mathsf{B}\,\mathsf{K}?\mathsf{m}} q_2' \in M_2$, $q_1 \xrightarrow{\mathsf{H}\,\mathsf{A}!\mathsf{m}} q_1' \in M_1$.*

The semi-direct composition of two systems takes all the machines of the participants in each system (with some channel renaming so to turn communications with H or K into communications with W) but the interface participants, which are replaced by the semi-direct gateway construction of their CFSMs.

Definition 4.2 (Semi-direct system composition). *Given two systems S_1 and S_2 with disjoint domain, two compatible roles $\mathsf{H} \in \mathrm{dom}(\mathsf{S}_1)$ and $\mathsf{K} \in \mathrm{dom}(\mathsf{S}_2)$, and a fresh role $\mathsf{W} \notin \mathrm{dom}(\mathsf{S}_1) \cup \mathrm{dom}(\mathsf{S}_2)$, the system*

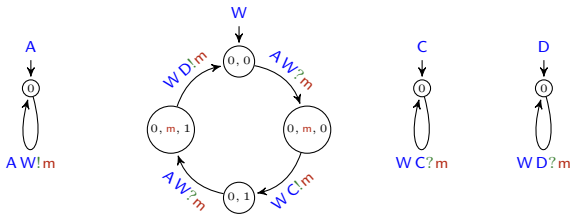
$$\mathsf{S}_1 \overset{\mathsf{H}\leftrightarrow\mathsf{K}}{\mathsf{w}} \mathsf{S}_2 : \mathsf{A} \mapsto \begin{cases} \mathsf{S}_i(\mathsf{A})[\mathsf{W}/\mathsf{H}][\mathsf{W}/\mathsf{K}], & \text{if } \mathsf{A} \in \mathrm{dom}(\mathsf{S}_i) \setminus \{\mathsf{H}, \mathsf{K}\} \text{ for } i \in \{1,2\} \\ \mathsf{sgw}(\mathsf{S}_1(\mathsf{H}), \mathsf{W}, \mathsf{S}_2(\mathsf{K})), & \text{if } \mathsf{A} = \mathsf{W} \end{cases}$$

is the W-composition of S_1 and S_2 with respect to H and K. *In the definition, the notation* $M[\mathsf{B}/\mathsf{A}]$ *denotes the machine obtained by replacing role* A *with* B *in all the labels of transitions in* M.

Note that since the gateway construction exploits the compatibility bisimilarity relation then the interface participants need to be compatible for the composition to make sense. This was not the case in the gateway construction in Section 3.

In the following simple example we show how the compatibility bisimilarity is exploited in the construction of a semi-direct composition.

Example 4.3. Let us take system S_1 with participants A and H and system S_2 with participants K, C and D as defined in Example 3.3. Participants H and K are trivially compatible. Then the following system



is the semi-direct composition $\mathsf{S}_1 \overset{\mathsf{H}\multimap\mathsf{K}}{\mathsf{W}} \mathsf{S}_2$. ◇

We now study systems obtained by semi-direct composition. As in the previous section, we will focus on preservation of deadlock-freedom.

Configurations of a composed system are projected on the two subsystems by taking only the states of their participants and the respective component of the states of the interfaces.

Definition 4.4 (Projection of configurations). *Given a configuration* $s \in \mathcal{R}([\![\mathsf{S}_1 \overset{\mathsf{H}\multimap\mathsf{K}}{\mathsf{W}} \mathsf{S}_2]\!])$, *the map* $s\|_i$, *for* $i \in \{1,2\}$, *defined as*

$$s\|_i : \mathsf{A} \mapsto \begin{cases} s(\mathsf{A}), & \text{if } \mathsf{A} \in \mathrm{dom}(\mathsf{S}_i) \setminus \{\mathsf{H},\mathsf{K}\} \\ q_1, & \text{if } \mathsf{A} = \mathsf{H} \text{ and either } s(\mathsf{W}) = (q_1, \mathsf{m}, q_2) \text{ or } s(\mathsf{W}) = (q_1, q_2) \\ q_2, & \text{if } \mathsf{A} = \mathsf{K} \text{ and either } s(\mathsf{W}) = (q_1, \mathsf{m}, q_2) \text{ or } s(\mathsf{W}) = (q_1, q_2) \end{cases}$$

is the projection of s on S_i.

As for the composition via gateways we define a notion of state projection to relate the states of the two systems.

Definition 4.5. *Let* $M = \mathsf{sgw}(M_{\mathsf{H}}, \mathsf{W}, M_{\mathsf{K}})$ *be a semi-direct gateway. The functions* $\mathsf{nofd}_i(\cdot)$, *where* $i \in \{1,2\}$, *on the states of* M *are defined as follows*

$$\mathsf{nofd}_i(q) = q_i \text{ if either } q = (q_1, q_2) \text{ or } (q_1, q_2) \xrightarrow{\mathsf{A\,W?m}} (q_1', \mathsf{m}, q_2') = q \text{ for some } \mathsf{m}, \mathsf{A}$$

We can now discuss the properties of composed systems.

Proposition 4.6. *Let* S_1 *and* S_2 *be two systems with disjoint domains and let* $\mathsf{H} \in \mathrm{dom}(\mathsf{S}_1)$ *and* $\mathsf{K} \in \mathrm{dom}(\mathsf{S}_2)$ *be two compatible roles. Then for each* $s \in \mathcal{R}(\llbracket \mathsf{S}_1 \overset{\mathsf{H}\multimap\mathsf{K}}{\mathsf{w}} \mathsf{S}_2 \rrbracket)$ *we have that*

i) $s\|_1 \in \mathcal{R}(\llbracket \mathsf{S}_1 \rrbracket)$, $s\|_2 \in \mathcal{R}(\llbracket \mathsf{S}_2 \rrbracket)$ *and* $\mathsf{nofd}_1(s(\mathsf{W})) \sim \mathsf{nofd}_2(s(\mathsf{W}))$;

ii) $s(\mathsf{A}) \xrightarrow{l} q$ *iff one of the following holds*

- *(a)* $\mathsf{A} \in \mathrm{dom}(\mathsf{S}_i) \setminus \{\mathsf{H}, \mathsf{K}\}$ *and* $s\|_i(\mathsf{A}) \xrightarrow{l} q$ *and* $\mathsf{W} \notin l$, *for* $i \in \{1, 2\}$;
- *(b)* $\mathsf{A} \in \mathrm{dom}(\mathsf{S}_1) \setminus \{\mathsf{H}\}$ *and* $l = \mathsf{A}\,\mathsf{W}!\mathsf{m}$ *and* $s\|_1(\mathsf{A}) \xrightarrow{\mathsf{A}\,\mathsf{H}!\mathsf{m}} q$;
- *(c)* $\mathsf{A} \in \mathrm{dom}(\mathsf{S}_1) \setminus \{\mathsf{H}\}$ *and* $l = \mathsf{W}\,\mathsf{A}?\mathsf{m}$ *and* $s\|_1(\mathsf{A}) \xrightarrow{\mathsf{H}\,\mathsf{A}?\mathsf{m}} q$;
- *(d)* $\mathsf{A} \in \mathrm{dom}(\mathsf{S}_2) \setminus \{\mathsf{K}\}$ *and* $l = \mathsf{A}\,\mathsf{W}!\mathsf{m}$ *and* $s\|_2(\mathsf{A}) \xrightarrow{\mathsf{A}\,\mathsf{K}!\mathsf{m}} q$;
- *(e)* $\mathsf{A} \in \mathrm{dom}(\mathsf{S}_2) \setminus \{\mathsf{K}\}$ *and* $l = \mathsf{W}\,\mathsf{A}?\mathsf{m}$ *and* $s\|_2(\mathsf{A}) \xrightarrow{\mathsf{K}\,\mathsf{A}?\mathsf{m}} q$;
- *(f)* $\mathsf{A} = \mathsf{W}$ *and* $s\|_1(\mathsf{H}) \xrightarrow{\mathsf{H}\,\mathsf{B}!\mathsf{m}} q_1$ *and* $s\|_2(\mathsf{K}) = q_2$ *and* $q = (q_1, q_2)$ *and* $l = \mathsf{W}\,\mathsf{B}!\mathsf{m}$;
- *(g)* $\mathsf{A} = \mathsf{W}$ *and* $s\|_1(\mathsf{H}) \xrightarrow{\mathsf{B}\,\mathsf{H}?\mathsf{m}} q_1$ *and* $s\|_2(\mathsf{K}) = q_2$ *and* $q = (q_1, \mathsf{m}, q_2)$ *and* $l = \mathsf{B}\,\mathsf{W}?\mathsf{m}$;
- *(h)* $\mathsf{A} = \mathsf{W}$ *and* $s\|_2(\mathsf{K}) \xrightarrow{\mathsf{K}\,\mathsf{B}!\mathsf{m}} q_2$ *and* $s\|_1(\mathsf{H}) = q_1$ *and* $q = (q_1, q_2)$ *and* $l = \mathsf{W}\,\mathsf{B}!\mathsf{m}$;
- *(i)* $\mathsf{A} = \mathsf{W}$ *and* $s\|_2(\mathsf{K}) \xrightarrow{\mathsf{B}\,\mathsf{K}?\mathsf{m}} q_2$ *and* $s\|_1(\mathsf{K}) = q_1$ *and* $q = (q_1, \mathsf{m}, q_2)$ *and* $l = \mathsf{B}\,\mathsf{W}?\mathsf{m}$;

Proof. The proof of (i) and (ii) is by simultaneous induction on the number of steps from the initial state. In the initial state (i) and (ii) hold by construction.

Let us consider the inductive case. We consider the following possible cases for the last transition.

- $s' \xrightarrow{\mathsf{A}\to\mathsf{B}:\,\mathsf{m}} s$ with $\mathsf{A}, \mathsf{B} \in \mathrm{dom}(\mathsf{S}_1)$ (the case $\mathsf{A}, \mathsf{B} \in \mathrm{dom}(\mathsf{S}_2)$ can be treated similarly).

  By definition of configuration transition, we have that $s'(\mathsf{A}) \xrightarrow{\mathsf{A}\,\mathsf{B}!\mathsf{m}} s(\mathsf{A})$ and $s'(\mathsf{B}) \xrightarrow{\mathsf{A}\,\mathsf{B}?\mathsf{m}} s(\mathsf{B})$ and $s'(\mathsf{C}) = s(\mathsf{C})$ for each $\mathsf{C} \in \mathrm{dom}(\mathsf{S}_1) \setminus \{\mathsf{A}, \mathsf{B}\}$. Now, by the induction hypothesis (ii), we have that $s'\|_1(\mathsf{A}) \xrightarrow{\mathsf{A}\,\mathsf{B}!\mathsf{m}} s(\mathsf{A}) = s\|_1(\mathsf{A})$ and $s'\|_1(\mathsf{B}) \xrightarrow{\mathsf{A}\,\mathsf{B}?\mathsf{m}} s(\mathsf{B}) = s\|_1(\mathsf{B})$ (where the two equalities can be inferred by definition of configuration projection, since $\mathsf{A}, \mathsf{B} \neq \mathsf{W}$). Hence we have that $s'\|_1 \xrightarrow{\mathsf{A}\to\mathsf{B}:\,\mathsf{m}} s\|_1$. Now, since by the induction hypothesis we have that $s'\|_1 \in \mathcal{R}(\llbracket \mathsf{S}_1 \rrbracket)$, we can infer that $s\|_1 \in \mathcal{R}(\llbracket \mathsf{S}_1 \rrbracket)$. We obtain, instead, $s\|_2 \in \mathcal{R}(\llbracket \mathsf{S}_2 \rrbracket)$ immediately by the induction hypothesis since, from $\mathsf{A}, \mathsf{B} \in \mathrm{dom}(\mathsf{S}_1)$ and definition of configuration projection we have that $s\|_2 = s'\|_2$. Also $\mathsf{nofd}_1(s(\mathsf{W})) \sim \mathsf{nofd}_2(s(\mathsf{W}))$ immediately follows from the induction hypothesis since $\mathsf{A}, \mathsf{B} \neq \mathsf{W}$ implies $s(\mathsf{W}) = s'(\mathsf{W})$. Regarding (ii), if $\mathsf{A} \neq \mathsf{W}$ then the same participant wants to take the same action thanks to (i), as desired. If $\mathsf{A} = \mathsf{W}$ is willing to communicate with some participant in S_1 then thanks to (i) and definition of semi-direct gateway, H is willing to do the same in $s\|_1$. Symmetrically, if $\mathsf{A} = \mathsf{W}$ is willing to communicate with some participant in S_2 then K is willing to do the same in $s\|_2$.

$s' \xrightarrow{\mathsf{A}\to\mathsf{W}:\,\mathsf{m}} s$ with $\mathsf{A} \in \mathrm{dom}(\mathsf{S}_1)$ (the case $\mathsf{A} \in \mathrm{dom}(\mathsf{S}_2)$ can be treated similarly). By definition of system transition, we have that $s'(\mathsf{A}) \xrightarrow{\mathsf{A}\,\mathsf{W}!\,\mathsf{m}} s(\mathsf{A})$ and $s'(\mathsf{W}) \xrightarrow{\mathsf{A}\,\mathsf{W}?\,\mathsf{m}} s(\mathsf{W})$ and $s'(\mathsf{C}) = s(\mathsf{C})$ for each $\mathsf{C} \in \mathrm{dom}(\mathsf{S}_1) \setminus \{\mathsf{A}\}$. Moreover, by the induction hypothesis, $s'\|_1 \in \mathcal{R}([\![\mathsf{S}_1]\!])$, $s'\|_2 \in \mathcal{R}([\![\mathsf{S}_2]\!])$ and $s'\|_1(\mathsf{H}) \sim s'\|_2(\mathsf{K})$. By definition of configuration projection and of semi-direct gateway construction we have that $s\|_2 = s'\|_2$, and hence we can immediately infer that $s\|_2 \in \mathcal{R}([\![\mathsf{S}_2]\!])$.
Now, by the induction hypothesis (ii), we have that $s'\|_1(\mathsf{A}) \xrightarrow{\mathsf{A}\,\mathsf{H}!\,\mathsf{m}} s(\mathsf{A})$ and $s'\|_1(\mathsf{H}) \xrightarrow{\mathsf{A}\,\mathsf{H}?\,\mathsf{m}} q_1$ and $s'\|_2(\mathsf{K}) = q_2$ where $s(\mathsf{W}) = (q_1, \mathsf{m}, q_2)$. Now, by definition of configuration projection, from $s(\mathsf{W}) = (q_1, \mathsf{m}, q_2)$ we obtain that $q_1 = s\|_1(\mathsf{W})$. So, by definition of configuration transition, we have that $s'\|_1 \xrightarrow{\mathsf{A}\to\mathsf{W}:\,\mathsf{m}} s\|_1$, and then $s\|_1 \in \mathcal{R}([\![\mathsf{S}_1]\!])$. For what concerns $\mathsf{nofd}_1(s(\mathsf{W})) \sim \mathsf{nofd}_2(s(\mathsf{W}))$, this is obtained by the induction hypothesis and by definition of $\mathsf{nofd}\_(\cdot)$ and of semi-direct gateway. Also, (ii) holds, as in the previous case, by (i) and definition of semi-direct gateway and of configuration projection.

$s' \xrightarrow{\mathsf{W}\to\mathsf{A}:\,\mathsf{m}} s$ with $\mathsf{A} \in \mathrm{dom}(\mathsf{S}_1)$ or $\mathsf{A} \in \mathrm{dom}(\mathsf{S}_2)$.
Similar to the previous case. □

We now give a definition of composability for semi-direct composition.

Definition 4.7 (Semi-direct (H,K)-composability). *Two systems S_1 and S_2 with disjoint domains are* semi-directly (H,K)-composable *if $\mathsf{H} \in \mathrm{dom}(\mathsf{S}_1)$ and $\mathsf{K} \in \mathrm{dom}(\mathsf{S}_2)$ are two compatible roles whose machines are ?!-deterministic and mixed-deterministic.*

Notice that semi-direct (H,K)-composability is strictly weaker than (H,K)-composability. In fact, whereas both require ?!-determinism, the former enables some mixed states whereas the latter completely forbids them.

It is easy to check that the counterexamples for deadlock-freedom preservation of Section 3 do hold also in case semi-directed gateways are used on (H,K)-composable systems. As before, this forces us to select interface roles which are sequential or which are in systems always willing to receive the messages they send. Before presenting the results we give an auxiliary lemma.

Lemma 4.8. *Let S_1 and S_2 be two semi-directly (H,K)-composable systems. Then for each configuration s of the composed system $\mathsf{S}_1 \overset{\mathsf{H}\rightleftharpoons\mathsf{K}}{\mathsf{w}} \mathsf{S}_2$ we have that:*

- *if $s(\mathsf{W}) = (q_1, q_2)$ then q_1, q_2 are in the compatibility bisimilarity;*
- *if $s(\mathsf{W}) = (q_1, \mathsf{m}, q_2)$ then $s(\mathsf{W})$ has a unique transition to and from a state of the form in the item above.*

Proof. By construction. Uniqueness relies on ?!- and mixed-determinism. □

Theorem 4.9 (Deadlock freedom for sequential interfaces). *Let S_1 and S_2 be two semi-directly (H,K)-composable and deadlock-free systems. If $\mathsf{S}_1(\mathsf{H})$ and $\mathsf{S}_2(\mathsf{K})$ are sequential, then the composed system $\mathsf{S}_1 \overset{\mathsf{H}\rightleftharpoons\mathsf{K}}{\mathsf{w}} \mathsf{S}_2$ is deadlock-free.*

Proof. We will show that if $\mathsf{S}_1 \overset{\mathsf{H}\leftrightarrows\mathsf{K}}{\mathsf{w}} \mathsf{S}_2$ has a deadlock then at least one of S_1 and S_2 has a deadlock as well.

First, Proposition 4.6(ii) immediately yields that for each configuration s of $\mathsf{S}_1 \overset{\mathsf{H}\leftrightarrows\mathsf{K}}{\mathsf{w}} \mathsf{S}_2$ if there is some participant A such that $s(\mathsf{A})$ has an outgoing transition, then for some participant B either $s\|_1(\mathsf{B})$ or $s\|_2(\mathsf{B})$ has an outgoing transition.

Now we show that if no transition is enabled in a configuration s of $\mathsf{S}_1 \overset{\mathsf{H}\leftrightarrows\mathsf{K}}{\mathsf{w}} \mathsf{S}_2$ then no transition is enabled in $s\|_1$ and $s\|_2$. We prove the contrapositive, showing that if there is an enabled transition in $s\|_1$ or in $s\|_2$ then there is a transition enabled in s as well. If the transition does not involve H, K this follows from Proposition 4.6. Let us now consider a transition involving H (the case of K is symmetric). If the transition is of the form $s\|_1 \xrightarrow{\mathsf{A}\longrightarrow\mathsf{H}\colon \mathsf{m}} \hat{s}$ and the state of W in s is a pair, by construction s can perform a transition $s \xrightarrow{\mathsf{A}\longrightarrow\mathsf{W}\colon \mathsf{m}} s'$ as desired. If the state of W in s is a triple then thanks to Lemma 4.8 and definition of semi-direct gateway, the previous state was in the compatibility bisimilarity with a state of K, which has not changed. Hence K is willing to take a transition and thanks to deadlock-freedom of S_2 we can infer that there is a transition enabled in $s\|_2$. From this we can deduce that there is a transition enabled in s too as shown above, but for the case in which the enabled transition is from K. In this last case thanks to sequentiality the transition towards H and the one from K are complementary, thus S_2 is ready to take the message from the gateway, hence the communication can trigger.

A similar reasoning applies in case the transition is of the form $s\|_1 \xrightarrow{\mathsf{H}\longrightarrow\mathsf{A}\colon \mathsf{m}} \hat{s}$.

Thus, if there is a deadlock configuration s in the composed system then either $s\|_1$ or $s\|_2$ are deadlocks against the hypothesis. □

Notice that S_1 and S_2 of Example 4.3 are deadlock-free and (H, K)-composable. Besides, both H and K are sequential. Deadlock-freedom of $\mathsf{S}_1 \overset{\mathsf{H}\leftrightarrows\mathsf{K}}{\mathsf{w}} \mathsf{S}_2$ can hence be inferred by the above result.

As done for the composition via gateways, we can extend the above result by dropping the sequentiality condition in presence of !live interfaces.

Theorem 4.10 (Deadlock freedom for !live interfaces). *Let* S_1 *and* S_2 *be two semi-directly* (H, K)*-composable and deadlock-free systems. Moreover, let* $\mathsf{S}_1(\mathsf{H})$ *and* $\mathsf{S}_2(\mathsf{K})$ *be* !*live, respectively, in* S_1 *and* S_2*. Then the composed system* $\mathsf{S}_1 \overset{\mathsf{H}\leftrightarrows\mathsf{K}}{\mathsf{w}} \mathsf{S}_2$ *is deadlock-free.*

Proof. The proof is similar to the one of Theorem 4.9. The only difference is that !liveness is used instead of sequentiality when showing that if there is an enabled transition in $s\|_1$ or in $s\|_2$ then there is a transition enabled in s as well. □

5 Related and Future Work

We have considered the synchronous composition of systems of CFSMs following the approach proposed in [3,4] for asynchronous composition. Quite surprisingly, enforcing that composition preserves deadlock freedom requires very strong

conditions on the interface roles, as shown by means of some examples. Indeed, we proved compositionality of deadlock freedom for sequential interface roles only. We hence complemented this result by showing that, if a deadlock occurs, it needs to be when the gateway tries to deliver a message to the other system.

We also discussed semi-direct composition, based on a unique gateway. Beyond sparing some communications, the conditions required to ensure compositionality of deadlock freedom using this second approach are slightly weaker.

While we only discussed deadlock freedom, the same reasonings can be applied to other behavioural properties such as lock freedom [19,18,6] and liveness [23,6].

The above approach to composition has also been discussed in [5], in the setting of systems of processes obtained by projecting well-formed global types [17]. This setting is far less wild than ours, since global types ensure that each send is matched by a receive. Thus, all the counterexamples we showed cannot happen and deadlock freedom is ensured in all typable systems. Thanks to these restrictions they were able to develop a more comprehensive theory, including direct composition, a notion of structural decomposition and notions of behavioural composition and decomposition. Also, they could use as compatibility a relation weaker than bisimilarity. Understanding whether such a theory can be amended to fit in our more general setting is an interesting item for future work.

Compositionality in the setting of global types has been also studied in [22]. There the compositionality mechanism is different since it relies on partial systems, while the approach we use allows one to compose systems which are designed as closed, by transforming some participants into gateways. On the other hand they are able to model ways of interaction more structured than having a single communication channel as in our case. Extending our approach to cope with the composition via multiple interfaces at the same time can be an interesting aim for future work and can contribute to match their expressive power.

A compositional approach for reactive components has been proposed in [12,25]. Composition is attained by means of a specified protocol regulating the communications between components that are supposed to produce results as soon as they get their inputs. Roughly speaking, this protocol represents the composition interface that rules out, among the communications of components, those not allowed in the composition. In this way, a component may be used in compositions under different protocols if its communications are compliant with (part of) the protocols. A difference with our approach is that the framework in [12,25], as common in session type approaches, requires the specification of a global type from which to derive local types to type check components in order to compose them.

Among the automata-based models in the literature, I/O automata [21], team automata [26], interface automata [15], and BIP [9] are perhaps the closest to communicating systems. In these models composition strategies based on some notion of compatibility have been proposed. However, these approaches differ from ours on a number of aspects.

First, the result of such a composition is a new automaton, not a system as in our case. Correspondingly, our notion of "interface" is more elaborated than

in the other models. Indeed, for us an interface is a pair of automata rather than sets of actions of a single automaton.

Second, such automata have a fixed interface, since they distinguish internal from external actions. Instead, we do not fix an explicit interface: the interface is decided in a *relative* fashion. This gives a high degree of flexibility; e.g., we could use as interface a CFSM H' when composing a system S with a system, say S', and a different CFSM H'' in S when composing it with another system S''.

As previously pointed out, and related to the previous observations, we could think of our approach as not been based on a notion of "open" systems. We compose *closed* systems by "opening them up" depending on their relative structures, namely on the fact that they possess compatible components.

Extensive studies about compositionality of interacting systems have been conducted in the context of the BIP model [9]. Composition in BIP happens through operators meant to mediate the behaviour of the connected components. The composition can alter the non-deterministic behaviour by suitable priority models. In [2,1] it is shown that, under mild hypothesis, priority models do not spoil deadlock freedom. This requires to compromise on expressiveness. Whether our conditions are expressible in some priority model is open and left for future work. BIP features multi-point synchronisations while CFSMs interactions are point-to-point. Very likely CFSMs can be encoded in BIP without priorities and one could use D-Finder [8] to detect deadlock of composed systems. However, our conditions on interfaces allows us to avoid such analysis.

In the present approach, the transformations generating the gateway(s) from the interface roles do not depend on the rest of the systems to be composed. Besides investigating relaxed notions of compatibility (in the style of [5]), it would also be worth considering the possibility of dropping the compatibility requirement altogether and developing methods to generate ad-hoc gateways (i.e., taking into account the other CFSMs of the two systems to be composed) that preserve deadlock freedom and communication properties in general by construction. In would also be worth investigating whether our approach can be extended to cope with types of message passing communications other than point-to-point, such as multicast [14], broadcast or many-to-many [11].

References

1. E. Baranov and S. Bliudze. Offer semantics: Achieving compositionality, flattening and full expressiveness for the glue operators in BIP. *Sci. Comput. Program.*, 109:2–35, 2015.
2. E. Baranov and S. Bliudze. Expressiveness of component-based frameworks: A study of the expressiveness of BIP. *Acta Informatica*, pages 1—40, 2019.
3. F. Barbanera, U. de'Liguoro, and R. Hennicker. Global types for open systems. In M. Bartoletti and S. Knight, editors, *ICE*, volume 279 of *EPTCS*, pages 4–20, 2018.
4. F. Barbanera, U. de'Liguoro, and R. Hennicker. Connecting open systems of communicating finite state machines. *JLAMP*, 109, 2019.
5. F. Barbanera, M. Dezani-Ciancaglini, I. Lanese, and E. Tuosto. Composition and decomposition of multiparty sessions. *JLAMP*, 2020. Submitted.

6. F. Barbanera, I. Lanese, and E. Tuosto. Choreography automata. In *COORDINATION*, volume 12134 of *LNCS*, pages 86–106. Springer, 2020.
7. S. Basu, T. Bultan, and M. Ouederni. Deciding choreography realizability. In *POPL*, pages 191–202, 2012.
8. S. Bensalem, A. Griesmayer, A. Legay, T. Nguyen, J. Sifakis, and R. Yan. D-finder 2: Towards efficient correctness of incremental design. In M. Bobaru, K. Havelund, G. J. Holzmann, and R. Joshi, editors, *NASA Formal Methods*, pages 453–458. Springer, 2011.
9. S. Bliudze and J. Sifakis. The algebra of connectors: structuring interaction in BIP. In *International conference on Embedded software*. ACM, Sept. 2020.
10. D. Brand and P. Zafiropulo. On communicating finite-state machines. *J. ACM*, 30(2):323–342, 1983.
11. R. Bruni, A. Corradini, F. Gadducci, H. C. Melgratti, U. Montanari, and E. Tuosto. Data-driven choreographies à la Klaim. In *Models, Languages, and Tools for Concurrent and Distributed Programming*, volume 11665 of *LNCS*, pages 170–190. Springer, 2019.
12. M. Carbone, F. Montesi, and H. T. Vieira. Choreographies for reactive programming. *CoRR*, abs/1801.08107, 2018. Available at `http://arxiv.org/abs/1801.08107`.
13. G. Cécé and A. Finkel. Verification of programs with half-duplex communication. *I&C*, 202(2):166–190, 2005.
14. M. Coppo, M. Dezani-Ciancaglini, N. Yoshida, and L. Padovani. Global progress for dynamically interleaved multiparty sessions. *Mathematical Structures in Computer Science*, 26(2):238–302, 2016.
15. L. De Alfaro and T. Henzinger. Interface automata. In *ACM SIGSOFT Software Engineering Notes*, volume 26(5), pages 109–120. ACM, 2001.
16. M. G. Gouda and C. Chang. Proving liveness for networks of communicating finite state machines. *ACM Trans. Program. Lang. Syst.*, 8(1):154–182, 1986.
17. H. Hüttel et al. Foundations of session types and behavioural contracts. *ACM Comput. Surv.*, 49(1):3:1–3:36, 2016.
18. N. Kobayashi. A partially deadlock-free typed process calculus. *ACM TOPLAS*, 20(2):436–482, 1998.
19. N. Kobayashi. Type-based information flow analysis for the pi-calculus. *Acta Informatica*, 42(4–5):291–347, 2005.
20. J. Lange, E. Tuosto, and N. Yoshida. From communicating machines to graphical choreographies. In *POPL*, pages 221–232. ACM, 2015.
21. N. Lynch and M. Tuttle. Hierarchical correctness proofs for distributed algorithms. In *ACM Symp. Principles of Distributed Computing*, pages 137–151. ACM, 1987.
22. F. Montesi and N. Yoshida. Compositional choreographies. In *CONCUR*, volume 8052 of *LNCS*, pages 425–439. Springer, 2013.
23. L. Padovani, V. T. Vasconcelos, and H. T. Vieira. Typing liveness in multiparty communicating systems. In *COORDINATION*, volume 8459 of *LNCS*, pages 147–162. Springer, 2014.
24. W. Peng and S. Purushothaman. Analysis of a class of communicating finite state machines. *Acta Inf.*, 29(6/7):499–522, 1992.
25. Z. Savanović, H. Vieira, and L. Galletta. A type language for message passing component-based systems. In *ICE*, EPTCS, 2020. To appear.
26. M. ter Beek and J. Kleijn. Team automata satisfying compositionality. In *FME 2003: Formal Methods*, pages 381–400. Springer, 2003.
27. E. Tuosto and R. Guanciale. Semantics of global view of choreographies. *JLAMP*, 95:17–40, 2018.