On Composing Communicating Systems

Franco Barbanera, Ivan Lanese, Emilio Tuosto

On Composing Communicating Systems *

Franco Barbanera
Dept. of Mathematics and Computer Science, University of Catania (Italy)

Ivan Lanese
Focus Team, University of Bologna/INRIA (Italy)

Emilio Tuosto
Gran Sasso Science Institute (Italy)

Communication is an essential element of modern software, yet programming and analysing communicating systems are difficult tasks. A reason for this difficulty is the lack of compositional mechanisms that preserve relevant communication properties.

This problem has been recently addressed for the well-known model of *communicating systems*, that is sets of components consisting of finite-state machines capable of exchanging messages. The main idea of this approach is to take two systems, select a participant from each of them, and derive from those participants a pair of coupled gateways connecting the two systems. More precisely, a message directed to one of the gateways is forwarded to the gateway in the other system, which sends it to the other system. It has been shown that, under some suitable *compatibility* conditions between gateways, this composition mechanism preserves deadlock freedom for asynchronous as well as symmetric synchronous communications (where sender and receiver play the same part in determining which message to exchange).

This paper considers the case of *asymmetric synchronous communications* where senders decide independently which message should be exchanged. We show here that preservation of lock freedom requires sequentiality of gateways, while this is not needed for preservation of either deadlock freedom or strong lock freedom.

1 Introduction

Communication is an essential constitutive element of modern software due to the fact that applications are increasingly developed in distributed architectures (e.g., service-oriented architectures, microservices, cloud, etc.). In practice, APIs and libraries featuring different communication mechanisms are available for practically any programming language. At a theoretical level, several models have been used to study interactions between systems (e.g., process algebras, transition systems, Petri nets, logical frameworks, etc.).

Reasoning and developing communicating systems are difficult endeavours. The so-called *business logic*, necessary to determine *what* has to be communicated, needs to be complemented with the so-called *application level protocol* specifying *how* information spreads across a system. Conceptual and programming errors may occur in the realisation of application level protocols. For instance, it may happen that some components in a system are prevented to communicate because all the expected partners terminated their execution (deadlock). Other typical errors occur when a system is not lock-free, that is when some components cannot progress because all their partners are perpetually involved in other

* 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 Progetto di Ateneo "Piaceri". The 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 Mariangiola Dezani for her support.

To appear in EPTCS.

communications. A source of these problems is when a system can evolve in different ways depending on some conditions and components have inconsistent "views" of the state of the system. If this happens, some components may reach a state no longer "compatible" with the state of their partners and therefore communications cannot take place as expected.

We illustrate these problems with some simple examples for deadlock freedom (similar examples may be given for lock freedom). Suppose we want to model a client-server system where clients' requests are acknowledged either with an answer or with "unknown" from servers. Due to its popularity, we choose CCS [16] to introduce this scenario, so take the following agents:

$$\mathsf{C} = \bar{\mathsf{r}}.\mathsf{a} + \bar{\mathsf{r}}.\mathsf{u} \qquad \text{and} \qquad \mathsf{D} = \mathsf{r}.\bar{\mathsf{a}} + \mathsf{r}.\bar{\mathsf{u}} \tag{1}$$

where ports r, a, and u are respectively used to commmunicate requests, answers, and unknowns. (Recall that in CCS $\mathsf{a}._$ and $\_ + \_$ represent respectively prefix and non-deterministic choice.) The common interpretation of agents in (1) is that x and $\bar{\mathsf{x}}$ respectively represent an input and an output on port x. It is a simple observation that the system $\mathsf{C} \mid \mathsf{D}$ where C and D run in parallel can evolve to e.g., the deadlock state $\mathsf{a} \mid \bar{\mathsf{u}}$ where each party is waiting for the other to progress. The problem is that the choice of what communication should happen after a request is taken independently by C and D instead of letting D to take the decision and drive C "on the right" branch. This is attempted in the next version:

$$\mathsf{C} = \bar{\mathsf{r}}.(\mathsf{a} + \mathsf{u}) \qquad \text{and} \qquad \mathsf{D} = \mathsf{r}.(\bar{\mathsf{a}} + \bar{\mathsf{u}}) \tag{2}$$

A key difference with the agents in (1) is that the server D in (2) decides what to reply to the client C, which becomes aware of the choice through the interaction with D after the request has been made. Let us now assume that the server D acts as a proxy to another server, say D'. When D cannot return an answer to the client it interacts with D' on port p. Answers are sent directly to the client if D' can compute them, otherwise D' returns an unknown on port u' to D which forwards it to the client. This is modelled by the agents

$$\mathsf{D} = \mathsf{r}.(\bar{\mathsf{a}} + \bar{\mathsf{p}}.\mathsf{u}'.\bar{\mathsf{u}}) \qquad \text{and} \qquad \mathsf{D}' = \mathsf{p}.(\bar{\mathsf{a}} + \bar{\mathsf{u}'}) \tag{3}$$

Note that this change is completely transparent to agent C, which in fact stays as in (2). It is now more difficult to ascertain if these choices may lead to a deadlock since the decision of D may involve also D'. Indeed, the parallel composition of agents in (3) may deadlock because, when C and D interact on port a, D' hangs on port p and, likewise, if C and D' interact on port a then D hangs on port u'.

A reason for this difficulty is that it is hard to define compositional mechanisms that preserve relevant communication properties such as deadlock or lock freedom. Recently, an approach to the composition of concurrent and distributed systems has been proposed in [2, 3] for the well-known model of systems of *communicating finite-state machines* (CFSMs) [11], that is sets of finite-state automata capable of exchanging messages. The compositional mechanism is based on the idea that two given systems, say S and S', are composed by transforming two CFSMs, say H in S and K in S', into "coupled forwarders". Basically, each message that H receives from a machine in S is forwarded to K and vice versa. It has been shown that, under suitable *compatibility* conditions between H and K, this composition mechanism preserves deadlock freedom for asynchronous as well as symmetric synchronous communications (where sender and receiver play the same part in determining which message to exchange). The compatibility condition identified in [2, 3] consists in exhibiting essentially dual behaviours: gateway H is able to receive a message whenever gateway K is willing to send one and vice versa. As observed in [4], a remarkable feature of such an approach is that it enables the composition of systems originally designed

as *closed* systems. As far as two compatible machines can be found, any two systems can be composed by transforming as hinted above the compatible machines.

The results in [2, 3] are developed in the asynchronous semantics of CFSMs. These results have been transferred in [5] to a setting where CFSMs communicate synchronously much like as the communication mechanisms considered for instance in process algebras like CCS, ACP, etc. This model assumes a perfect symmetry between sender and receiver in synchronous communications. Let us again discuss this with an example. Consider the agents

$$\mathsf{T} = \bar{\mathsf{a}}.\mathsf{P} + \bar{\mathsf{b}}.\mathsf{Q} \qquad \text{and} \qquad \mathsf{R} = \mathsf{a}.\mathsf{P}' + \mathsf{b}.\mathsf{Q}' \tag{4}$$

According to the standard semantics of CCS [16], system $(\mathsf{T} \mid \mathsf{R}) \setminus \{\mathsf{a}, \mathsf{b}\}$ has two possible evolutions:

$$(\mathsf{T} \mid \mathsf{R}) \setminus \{\mathsf{a}, \mathsf{b}\} \xrightarrow{\tau} \mathsf{P} \mid \mathsf{P}' \qquad \text{and} \qquad (\mathsf{T} \mid \mathsf{R}) \setminus \{\mathsf{a}, \mathsf{b}\} \xrightarrow{\tau} \mathsf{Q} \mid \mathsf{Q}'$$

namely, either both T and R opt for the "leftmost" branch (synchronising on a) or they both choose the "rightmost" one (synchronising on b). (Recall that in CCS $\_ \setminus X$ is the hiding of ports in the set X and that τ represents an internal action.) This means that, the resolution of the choice is implicit in the communication mechanism: a branch is taken as soon as T and R synchronise on the corresponding port. Intuitively, no distinction is made between sender and receiver (formally they are indeed interchangeable); this implies that the communication mechanism is at the very core of choice resolution [5].

Interestingly, for synchronous communications, an alternative interpretation is actually possible where this perfect symmetry is not assumed so that sender and receiver play different roles in choice resolution while still relying on synchronous communication. Let us explain this interpretation using again CCS. Consider a variant of CCS where outputs must be enabled before being fired. One could formally specify that with the following reduction rules:

$$\bar{\mathsf{a}}.\mathsf{P} + \mathsf{P}' \xrightarrow{\tau} \bar{\bar{\mathsf{a}}}.\mathsf{P} \qquad \text{and} \qquad \bar{\bar{\mathsf{a}}}.\mathsf{P} \mid (\mathsf{a}.\mathsf{Q} + \mathsf{Q}') \xrightarrow{\tau} \mathsf{P} \mid \mathsf{Q} \tag{5}$$

whereby the leftmost rule *chooses* one of the possible outputs of the sender (the chosen output is marked by the double bar in our notation) and the rightmost rule actually synchronises sender and receiver. This semantics is an abstract model of *asymmetric* communications (used e.g., in [7, 17]), where silent steps taken using the left rule model some internal computation of the sender to decide what to communicate to the partner. In other words, now the choice is entirely resolved on "one side" while the communication is a mere interaction of complementary actions, the output and the input. This asymmetry, at the core of asynchronous communication, can therefore also carry for synchronous communication.

It is worth observing that asymmetric communications abstract a rather common programming pattern where sending components may choose the output to execute *depending* on some internal computation. For instance, elaborating on the proxy scenario in (3), D could decide to directly send unknowns to normal clients while reserving the use of D' only for "privileged" clients.

Contributions. This paper transfers the composition by gateway mechanism of [2, 3] to the case of asymmetric synchronous communication of CFSMs. The main technical results are that, in the asymmetric case, gateway composition

- preserves deadlock freedom (as well as a strong version of lock freedom) provided that systems are composable (the relation of compatibility – one of the requirements for systems to be composable – in the present paper is less restrictive than the one used in [5]);

- preserves lock freedom if systems are composable and gateways are *sequential*, namely each state has at most one outgoing transition.

Interestingly, preservation of deadlock freedom can be guaranteed under milder conditions than in the symmetric case. In fact, sequentiality of gateways is not necessary to preserve deadlock freedom in the asymmetric case, while it is in the symmetric one.

Structure of the paper. Section 2 introduces systems of (asymmetric synchronous) CFSMs, related notions and communication properties. Composition by gateways is introduced and discussed in Section 3 together with the compatibility relation. Section 4 discusses the issues that prevent gateway composition to preserve communication properties. Section 5 is devoted to the preservation of communication properties. Conclusions, related and future work are discussed in Section 6.

2 Background

Communicating Finite State Machines (CFSMs) [11] are Finite State Automata (FSAs) where transitions are labelled by communications. We recall basic notions on FSAs.

A *finite state automaton* (FSA) is a tuple $A = \langle \mathscr{S}, q_0, \mathscr{L}, \rightarrow \rangle$ where

- $\mathscr{S}$ is a finite set of states (ranged over by lowercase italic Latin letters);
- $q_0 \in \mathscr{S}$ is the *initial state*;
- $\mathscr{L}$ is a finite set of labels
- $\rightarrow \subseteq \mathscr{S} \times (\mathscr{L} \cup \{\tau\}) \times \mathscr{S}$ is a set of transitions.

Hereafter, we let λ range over $\mathscr{L} \cup \{\tau\}$ when it is immaterial to specify the set of labels or it is understood. We use the usual notation $q_1 \xrightarrow{\lambda} q_2$ for the transition $(q_1, \lambda, q_2) \in \rightarrow$, and $q_1 \rightarrow q_2$ when there exists a label λ such that $q_1 \xrightarrow{\lambda} q_2$. Let $\_\cdot\_$ be the concatenation operation on labels and write $p \xrightarrow{\pi} q$ where $\pi = \lambda_1 \cdot \lambda_2 \cdot \ldots \cdot \lambda_n$ whenever $p \xrightarrow{\lambda_1} p_1 \xrightarrow{\lambda_2} \ldots \xrightarrow{\lambda_n} p_n = q$. We let $\pi, \psi, \ldots$ range over $\mathscr{L}^\star$ (i.e., sequences of labels) and define the set of *reachable states in A from q* as

$$\mathscr{R}(A,q) = \{\, p \mid \text{there is } \pi \in \mathscr{L}^\star \text{ such that } q \xrightarrow{\pi} p \,\}$$

The set of *reachable states in A* is $\mathscr{R}(A) = \mathscr{R}(A, q_0)$. For succinctness, $q \xrightarrow{\lambda} q' \in A$ means that the transition belongs to (the set of transitions of) A; likewise, $q \in A$ means that q belongs to the states of A. We say that $q \xrightarrow{\lambda} q'$ is an *outgoing* (resp. *incoming*) transition of q (resp. q'). Since we use FSAs to formalise communicating systems, accepting states are disregarded (as also done in [11]).

We now define systems of CFSMs, by adapting the definitions in [11] to our context. Let $\mathfrak{P}$ be a set of *participants* (or *roles*, ranged over by A, B, etc.) and $\mathscr{M}$ a set of *messages* (ranged over by m, n, etc.). We take $\mathfrak{P}$ and $\mathscr{M}$ disjoint. An *output label* is written as A B!m and represents the willingness of A to send message m to B; likewise, an *input label* is written as A B?m and represents the willingness of B to receive message m from A. The *subjects* of an output label A B!m and of an input label A B?m are A and B, respectively.

Definition 2.1 (CFSMs)**.** *A* communicating finite-state machine *(CFSM) is an FSA M with labels in the set* $\mathscr{L}_{act} \cup \{\tau\}$*, where*

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

and, for any transition $p \xrightarrow{\lambda} q$*,*

- *if* λ *is an output label then* $p \neq q$ *and* p *has exactly one incoming transition, and such transition is labelled by* τ*;*
- *if* $\lambda = \tau$ *then* $p \neq q$ *and* q *has exactly one outgoing transition, and such transition is labelled by an output label.*

A state of M *is*

- terminal*, if it has no outgoing transition; we define* $\mathsf{T}(M) = \{\, p \in M \mid p \text{ is terminal in } M \,\}$
- sending*, if it is not terminal and all its outgoing transitions have output labels*
- receiving*, if it is not terminal and all its outgoing transitions have input labels*
- mixed*, if it has a silent outgoing transition and an outgoing transition with an input label.*

A CFSM is A-local *if all its non* τ *transitions have subject* A.

Unlike in [5], the transitions of our CFSMs can also be labelled by the silent action τ. Definition 2.1 can be looked at as the CFSM counterpart of the τC contracts described in [6]. Imposing the no-mixed state condition on our CFSM, turns them into the communicating automata counterpart of the processes (contracts) called "session behaviours"[1] in e.g., [10, 1, 7]. These processes are in turn the process counterpart of (binary) session types [13]. As we shall see below (and also shown in [3] and [5]), the absence of mixed states is necessary in order to get the preservation of properties by composition. As a matter of fact, we could drop the conditions related to τ-transitions in case a transition like $p \xrightarrow{\mathsf{X}\mathsf{Y}!\mathsf{z}} q$ is the only outgoing transition from p, namely when no actual choice of output actions is possible in p. We however prefer to avoid this distinction for several reasons:

- firstly, our uniform treatment of transitions allows us to immediately adapt definitions in a more abstract setting;
- secondly, uniformity allows us to simplify some technicalities.

Said that, all proofs in the present paper could easily be adapted to the above mentioned alternative definition of CFSM.

Definition 2.2 (Communicating systems)**.** *A* (communicating) system over $\mathscr{P}$ *is a map* $\mathsf{S} = (M_{\mathsf{A}})_{\mathsf{A} \in \mathscr{P}}$ *assigning an* A*-local CFSM* M_{A} *to each participant* $\mathsf{A} \in \mathscr{P}$ *where* $\mathscr{P} \subseteq \mathfrak{P}$ *is finite and any participant occurring in a transition of* M_{A} *is in* $\mathscr{P}$.

Note that Definition 2.2 requires that any input or output label does refer to participants belonging to the system itself. In other words, Definition 2.2 restricts to *closed* systems.

We now define the synchronous semantics of communicating systems, which is itself an FSA (differently from the asynchronous case, where the set of states can be infinite). Hereafter, 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.

Definition 2.3 (Asymmetric synchronisations)**.** *Let* S *be a communicating system. A* 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 asymmetric synchronisations *of* S *is the FSA*

$$[\![\mathsf{S}]\!] = \langle \mathscr{S}, s_0, \mathscr{L}_{int} \cup \{\, \tau \,\}, \rightarrow \rangle \qquad \textit{where}$$

- $\mathscr{S}$ *is the set of synchronous configurations of* S*, as defined above;*

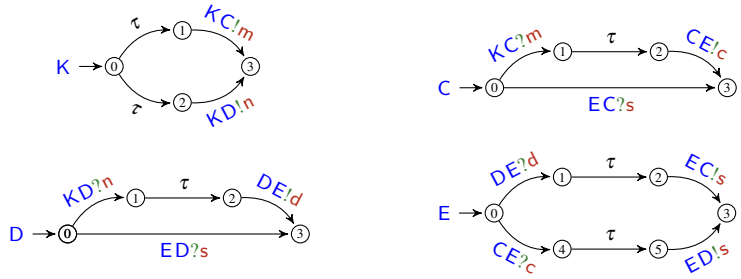
[1] Actually different variations of this name are used in the listed references.

- $s_0 = (q_{0\mathsf{A}})_{\mathsf{A}\in\mathrm{dom}(\mathsf{S})} \in \mathscr{S}$ *is the* initial *configuration where, for each* $\mathsf{A} \in \mathrm{dom}(\mathsf{S})$, $q_{0\mathsf{A}}$ *is the initial state of* $\mathsf{S}(\mathsf{A})$*;*
- $\mathscr{L}_{int} = \{\mathsf{A}{\to}\mathsf{B}\colon \mathsf{m} \mid \mathsf{A} \neq \mathsf{B} \in \mathfrak{P} \text{ and } \mathsf{m} \in \mathscr{M}\}$ *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* $s(\mathsf{B}) \xrightarrow{\mathsf{A}\mathsf{B}?\mathsf{m}} q' \in \mathsf{S}(\mathsf{B})$*;*
- $s \xrightarrow{\tau} s[\mathsf{A} \mapsto q] \in [\![\mathsf{S}]\!]$ *if* $s(\mathsf{A}) \xrightarrow{\tau} q \in \mathsf{S}(\mathsf{A})$*;*

Configuration s enables A in S *if* $s(\mathsf{A})$ *has at least an outgoing transition.*

As expected, an interaction $\mathsf{A}{\to}\mathsf{B}\colon \mathsf{m}$ occurs when A performs an output $\mathsf{A}\mathsf{B}!\mathsf{m}$ (which has been previously chosen) and B the corresponding input $\mathsf{A}\mathsf{B}?\mathsf{m}$.

Example 2.4. *Let us consider the communicating system* $\mathsf{S} = (M_{\mathsf{X}})_{\mathsf{X}\in\{\mathsf{K},\mathsf{C},\mathsf{D},\mathsf{E}\}}$*, where*



A sequence of transitions of $[\![\mathsf{S}]\!]$ *out of* s_0 *is, according to Definition 2.3,*

$$\begin{aligned} s_0 = (0_{\mathsf{K}}, 0_{\mathsf{C}}, 0_{\mathsf{D}}, 0_{\mathsf{E}}) &\xrightarrow{\tau} (1_{\mathsf{K}}, 0_{\mathsf{C}}, 0_{\mathsf{D}}, 0_{\mathsf{E}}) \xrightarrow{\mathsf{K}\to\mathsf{C}\colon \mathsf{m}} (3_{\mathsf{K}}, 1_{\mathsf{C}}, 0_{\mathsf{D}}, 0_{\mathsf{E}}) \\ &\xrightarrow{\tau} (3_{\mathsf{K}}, 2_{\mathsf{C}}, 0_{\mathsf{D}}, 0_{\mathsf{E}}) \xrightarrow{\mathsf{C}\to\mathsf{E}\colon \mathsf{c}} (3_{\mathsf{K}}, 3_{\mathsf{C}}, 0_{\mathsf{D}}, 4_{\mathsf{E}}) \\ &\xrightarrow{\tau} (3_{\mathsf{K}}, 3_{\mathsf{C}}, 0_{\mathsf{D}}, 5_{\mathsf{E}}) \xrightarrow{\mathsf{E}\to\mathsf{D}\colon \mathsf{s}} (3_{\mathsf{K}}, 3_{\mathsf{C}}, 3_{\mathsf{D}}, 3_{\mathsf{E}}) \end{aligned}$$

◇

The symmetric synchronisation in [5] for systems without τ-transitions can be readily obtained from the above definition by disregarding the clause for the τ-transitions.

In the following, $\mathsf{ptp}(\tau) = \emptyset$ and $\mathsf{ptp}(\mathsf{A}{\to}\mathsf{B}\colon \mathsf{m}) = \mathsf{ptp}(\mathsf{A}\mathsf{B}!\mathsf{m}) = \mathsf{ptp}(\mathsf{A}\mathsf{B}?\mathsf{m}) = \{\mathsf{A},\mathsf{B}\}$ and, for a sequence $\pi = \lambda_1 \cdot \cdots \cdot \lambda_n$, we let $\mathsf{ptp}(\pi) = \cup_{1\le i\le n}\mathsf{ptp}(\lambda_i)$.

As discussed in Section 1, we shall study the preservation of communication properties under composition. We shall consider the following relevant properties: deadlock freedom, lock freedom and strong lock freedom. The definitions below adapt the ones in [12] to a synchronous setting (as done also in [15, 18, 5]).

Definition 2.5 (Communication properties)**.** *Let* S *be a communicating system on* $\mathscr{P}$*. We say that a participant* $\mathsf{A} \in \mathscr{P}$ *is* involved *in a run* $s \xrightarrow{\lambda_1} s_1 \ldots \xrightarrow{\lambda_n} s_n$ *of* S *if there is* $1 \le i \le n$ *such that either* $\mathsf{A} \in \mathit{ptp}(\lambda_i)$ *or* $\lambda_i = \tau$*,* $s_i(\mathsf{A}) \xrightarrow{\tau} q$ *in* $\mathsf{S}(\mathsf{A})$*, and* $s_{i+1} = s_i[\mathsf{A} \mapsto q]$*.*

Deadlock freedom *A configuration* $s \in \mathscr{R}([\![\mathsf{S}]\!])$ *is a* deadlock *if*

- *s has no outgoing transitions in* $[\![\mathsf{S}]\!]$ *and*
- *there exists* $\mathsf{A} \in \mathscr{P}$ *such that* $s(\mathsf{A})$ *enables* A *in* S*.*

A system is deadlock-free *if none of its configurations is a deadlock.*

Lock freedom *Let* $\mathsf{A} \in \mathscr{P}$*. A configuration* $s \in \mathscr{R}([\![\mathsf{S}]\!])$ *is a* lock *for* A *if*

- $s(\mathsf{A})$ *has outgoing transitions; and*
- A *is not involved in any run from* s.

A system is lock-free *if none of its configurations is a lock for any of its participants.*

Strong lock freedom *System* S *is* strongly lock-free *for* $\mathsf{A} \in \mathscr{P}$ *if for each* $s \in \mathscr{R}(\llbracket \mathsf{S} \rrbracket)$ *enabling* A *in* S *then* A *is involved in all maximal sequences from* s.

A system is strongly lock free *if it is strongly lock free for each of its participants.*

Proposition 2.6. *1. Lock-freedom implies deadlock-freedom;*

2. Strong lock freedom implies lock freedom.

Example 2.7. *Let us consider the system* S *of Example 2.4. The only other maximal transition sequence in* $\llbracket \mathsf{S} \rrbracket$ *out of* s_0*, besides the one described in Example 2.4, is*

$$\begin{array}{rcl} s_0 = (0_\mathsf{K}, 0_\mathsf{C}, 0_\mathsf{D}, 0_\mathsf{E}) & \xrightarrow{\tau} & (2_\mathsf{K}, 0_\mathsf{C}, 0_\mathsf{D}, 0_\mathsf{E}) \xrightarrow{\mathsf{K}\to\mathsf{D}:\mathsf{n}} (3_\mathsf{K}, 0_\mathsf{C}, 1_\mathsf{D}, 0_\mathsf{E}) \\ & \xrightarrow{\tau} & (3_\mathsf{K}, 0_\mathsf{C}, 2_\mathsf{D}, 0_\mathsf{E}) \xrightarrow{\mathsf{D}\to\mathsf{E}:\mathsf{d}} (3_\mathsf{K}, 0_\mathsf{C}, 3_\mathsf{D}, 1_\mathsf{E}) \\ & \xrightarrow{\tau} & (3_\mathsf{K}, 0_\mathsf{C}, 3_\mathsf{D}, 2_\mathsf{E}) \xrightarrow{\mathsf{E}\to\mathsf{C}:\mathsf{s}} (3_\mathsf{K}, 3_\mathsf{C}, 3_\mathsf{D}, 3_\mathsf{E}) \end{array}$$

These two sequences are both maximal and contain all the elements of $\mathscr{R}(\llbracket \mathsf{S} \rrbracket)$*. By the above observations it is possible to check* S *to be strongly lock free.* ◇

3 Composition via Gateways

This section discusses composition of systems of CFSMs via gateways, as introduced in [2, 3], and studies its properties under asymmetric synchronisation. 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.

3.1 Building gateways

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 in* M

- *each pair of consecutive transitions* $p \xrightarrow{\tau} q \xrightarrow{\mathsf{HA!m}} r$ *with*

$$p \xrightarrow{\mathsf{KH?m}} p' \xrightarrow{\tau} q \xrightarrow{\mathsf{HA!m}} r \qquad \text{for some fresh state } p' \tag{6}$$

- *each transition* $p \xrightarrow{\mathsf{AH?m}} r$ *with*

$$p \xrightarrow{\mathsf{AH?m}} p' \xrightarrow{\tau} p'' \xrightarrow{\mathsf{HK!m}} r \qquad \text{for some fresh states } p' \text{ and } p'' \tag{7}$$

We shall call external *the states like p and r and* internal *the states like p', p'' and q.*

Note that gateways execute "segments" of the form described in (6) and (7) in the above definition. Also, by very construction, we have the following

Fact 3.2. *Given a* H*-local CFSM M and a participant* K*, each state of* $\mathsf{gw}(M,\mathsf{K})$ *has at most one incoming or outgoing τ transition.*

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.

Given two functions f and g such that $\mathrm{dom}(f)\cap\mathrm{dom}(g)=\emptyset$, we let $f+g$ denote the function behaving as function f on $\mathrm{dom}(f)$ and as function g on $\mathrm{dom}(g)$.

Definition 3.3 (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*

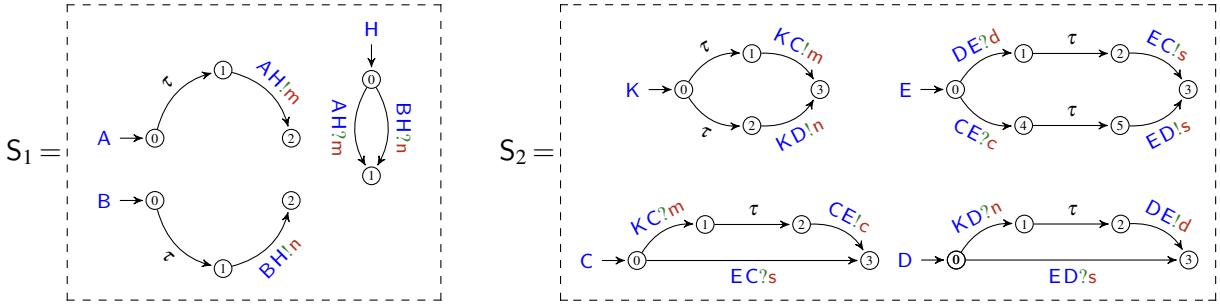
$$\mathsf{S}_1{}^{\mathsf{H}\leftrightarrow\mathsf{K}}\,\mathsf{S}_2=\mathsf{S}_1[\mathsf{H}\mapsto\mathsf{gw}(\mathsf{S}_1(\mathsf{H}),\mathsf{K})]+\mathsf{S}_2[\mathsf{K}\mapsto\mathsf{gw}(\mathsf{S}_2(\mathsf{K}),\mathsf{H})]$$

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

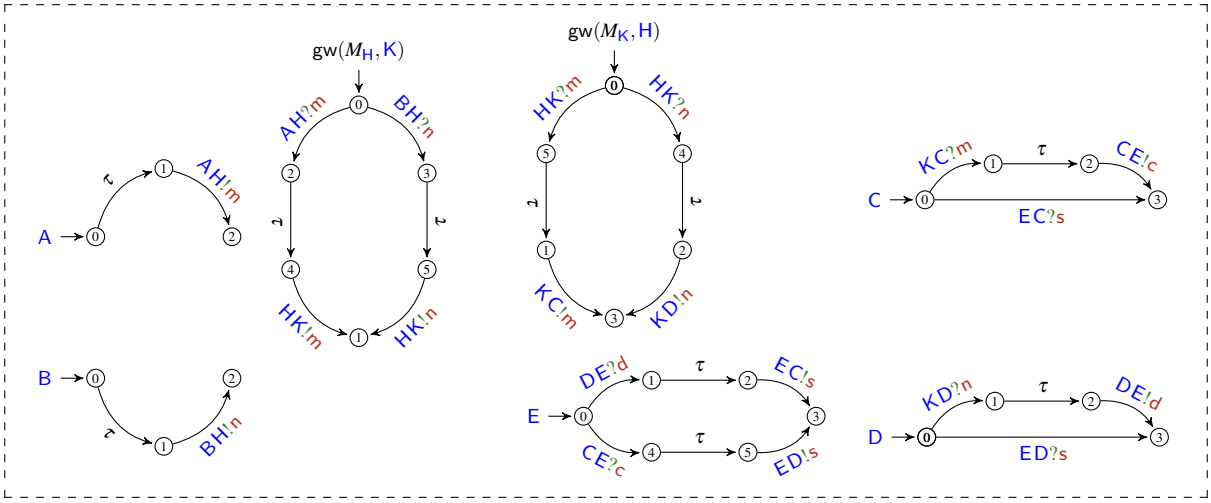
We remark again that, by the above approach for composition, we do not actually need to formalise the notion of *open* system. In fact any closed system can be looked at as open by choosing (according to the current needs) two suitable participants in the "to-be-connected" systems and transforming them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

We also note that the notion of composition above is structural: a corresponding notion of behavioural composition has been studied in [4] in a context of multiparty session types [14], that is with symmetric synchronous interactions.

Example 3.4. *Let us take the following two communicating systems.*



The system $\mathsf{S}_1{}^{\mathsf{H}\leftrightarrow\mathsf{K}}\,\mathsf{S}_2$ *is*



Note that the CFSMs A, B, C, D, *and* E *remain unchanged.* ◇

3.2 Compatibility

A few simple auxiliary notions are useful. Let $\mathscr{L}_{\text{i/o}} = \{\, ?\mathsf{m}, !\mathsf{m} \mid \mathsf{m} \in \mathscr{M} \,\}$ and define the functions

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

by the following clauses

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

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

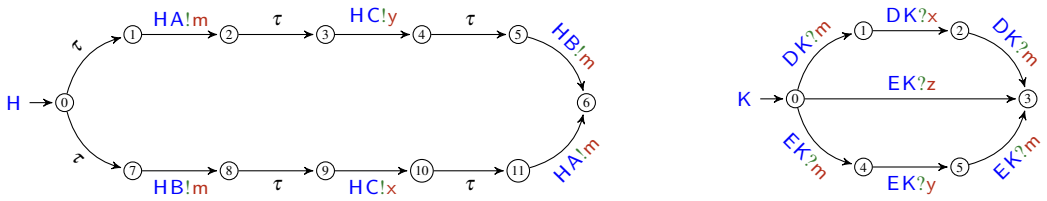
Informally, two CFSMs M_1 and M_2 are *compatible* if each output of M_1 has a corresponding input in M_2 and vice versa once the identities of communicating partners are blurred away.

Definition 3.5 (Compatibility)**.** *Let M and M' be two FSAs on $\mathscr{L}_{\text{i/o}}$. An* io-correspondence *is a relation R between states of M and those of M' such that whenever $(q, q') \in R$:*

- $q \in \mathsf{T}(M)$ *if, and only if,* $q' \in \mathsf{T}(M')$ *(cf. Definition 2.1)*
- *if* $q \xrightarrow{!\mathsf{m}} r \in M$ *then there is* $q' \xrightarrow{?\mathsf{m}} r' \in M'$ *such that* $(r, r') \in R$
- *if* $q' \xrightarrow{!\mathsf{m}} r' \in M'$ *then there is* $q \xrightarrow{?\mathsf{m}} r \in M$ *such that* $(r, r') \in R$
- *if* $q \xrightarrow{\tau} r \in M$ *then* $(r, q') \in R$
- *if* $q' \xrightarrow{\tau} r' \in M'$ *then* $(q, r') \in R$

Two CFSMs M and M' are compatible *(in symbols $M \asymp M'$) if there is an io-correspondence relating the initial states of* $\mathsf{io}(M_1)$ *and* $\mathsf{io}(M')$.

Example 3.6 (Compatibility). *The machines* H *and* K *of Example 3.4 are compatible. For a more complex example, consider the following CFSMs*



The above H *and* K *are compatible. Apart for* τ *actions preceding them,* H *can only perform output actions, whereas* K *can only perform input actions. By disregarding the names of the receivers in the actions of* H*, and of the senders in those of* K*, any output action after its corresponding* τ *can find a matching input in* K*. The vice versa does not hold, since none of the possible output actions that can occur after a* τ *from* 0 *(i.e. the outputs from* 1 *and* 7 *in* H*) can actually match the input action* EK?z *from* 0 *in* K*. Such a possibility is in fact allowed by our definition of compatibility.* ◇

Definition 3.5 transfers the notion of compatibility given in [4] for processes in multiparty sessions. Also, Definition 3.5 differs from the notions of compatibility in [5] and in [2, 3] which are defined as bisimulations and do not involve τ-transitions.

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

1. ?-deterministic *if* $p \xrightarrow{\mathsf{XA?m}} q$ *and* $p \xrightarrow{\mathsf{YA?m}} r \in M$ *implies* $q = r$*;*
2. !-deterministic *if* $p \xrightarrow{\tau} \xrightarrow{\mathsf{AX!m}} q$ *and* $p \xrightarrow{\tau} \xrightarrow{\mathsf{AY!m}} r \in M$ *implies* $q = r$*;*
3. ?!-deterministic *if it is both ?-deterministic and !-deterministic;*

A non-terminal state $q \in M$ is *asymmetric sending* (resp. *receiving*) if all its outgoing transitions have τ (resp. receiving) labels; q is a *asymmetric mixed* state if it is neither asymmetric sending nor receiving.

Example 3.8. *Machine* H *and* K *in Example 3.6 are, respectively non !-deterministic and non ?-deterministic. In particular, conditions (2) and (1) of Definition 3.7 fail for, respectively, state* 0 *of* H *and state* 0 *of* K*.*

We require a stronger condition then compatibity for two systems to be composable.

Definition 3.9 ((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 ?!-deterministic machines with no asymmetric mixed states.*

4 Composition Related Issues

It is known that under symmetric synchronisation composition spoils deadlock-freedom; this is shown by the example below, borrowed from [5].

Example 4.1 (Deadlock-freedom preservation fails under symmetric synchronisation). *Take the following systems*

Clearly, S_1 and S_2 are (H,K)-composable and deadlock-free, yet their composition $\mathsf{S} = \mathsf{S}_1 \overset{\mathsf{H}\leftrightarrow\mathsf{K}}{} \mathsf{S}_2$ has a deadlock. In fact, when the gateway for K receives m, it tries to synchronise with participant B on message m while B is waiting only for x. For S_2 in isolation, this is not a deadlock, since B and K synchronise on x under the symmetric semantics. ◇

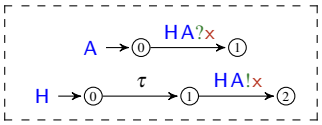
Notice that the counterexample of Example 4.1 does not apply in an asynchronous setting. Indeed, the second system could deadlock due to the fact that K could send m without synchronising with B. Likewise, the counterexample of Example 4.1 does not apply in our asymmetric setting. Even if communication is still synchronous, the τ-transitions introduced to resolve internal choices (i.e., those prefixing outputs) allow S_2 to reach a deadlock configuration by choosing the τ-transition leading to the output $\mathsf{K}\,\mathsf{B}!\mathsf{m}$.

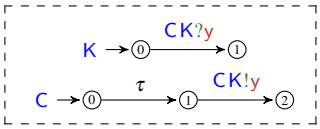
Now, one may think that analogously to what happens in [2, 3], if two systems are (H,K)-composable and deadlock-free then their composition is deadlock-free too.

In Section 5 we shall prove that in our setting lock-freedom is preserved by composition, without any further condition beside (H,K)-composability. Before doing that, we give examples showing the necessity of our conditions for deadlock freedom preservation.

Let us begin with compatibility. Properties cannot be preserved under composition without compatibility, as shown in the next example.

Example 4.2 (Lack of compatibility spoils deadlock freedom preservation)**.** *Let us consider the following communicating systems.*

$\mathsf{S}_1 =$ 

and $\mathsf{S}_2 =$ 

These systems are trivially deadlock free. However, H and K are not compatible, since there is no corresponding input in K for the output from H. The composition of S_1 and S_2 via H and K yields

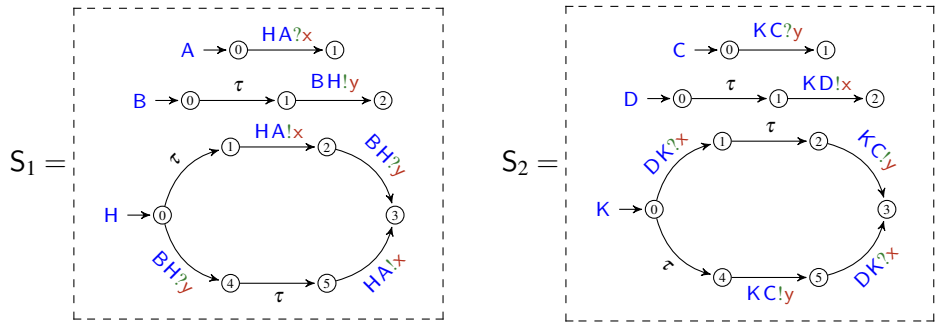
Starting from the initial configuration of $\mathsf{S}_1 \overset{\mathsf{H}\leftrightarrow\mathsf{K}}{} \mathsf{S}_2$, the following transitions are possible in $[\![\mathsf{S}_1 \overset{\mathsf{H}\leftrightarrow\mathsf{K}}{} \mathsf{S}_2]\!]$

$$(0_\mathsf{A},0_\mathsf{H},0_\mathsf{K},0_\mathsf{C}) \xrightarrow{\tau} (0_\mathsf{A},0_\mathsf{H},0_\mathsf{K},1_\mathsf{C}) \xrightarrow{\mathsf{C}\to\mathsf{K}:\,\mathsf{y}} (0_\mathsf{A},0_\mathsf{H},2_\mathsf{K},2_\mathsf{C}) \xrightarrow{\tau} (0_\mathsf{A},0_\mathsf{H},3_\mathsf{K},2_\mathsf{C}) \not\to$$

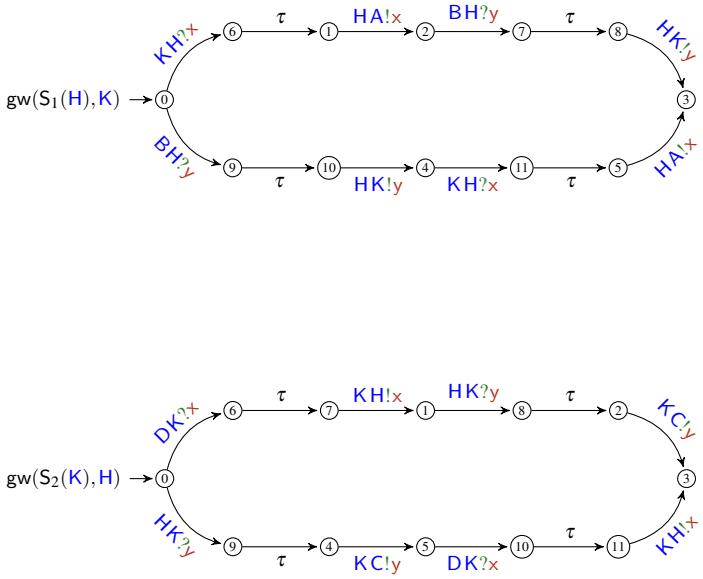
where $(0_\mathsf{A},0_\mathsf{H},3_\mathsf{K},2_\mathsf{C})$ is a deadlock configuration for $[\![\mathsf{S}_1 \overset{\mathsf{H}\leftrightarrow\mathsf{K}}{} \mathsf{S}_2]\!]$ since K wishes to send y to H, which is instead waiting for message x. ◇

The following example casts in our setting an example given in [3] for the asynchronous semantics; this example illustrates that asymmetric mixed states must be avoided to preserve properties under composition.

Example 4.3 (Asymmetric mixed-states spoil deadlock freedom preservation)**.** *Let* S_1 *and* S_2 *be*



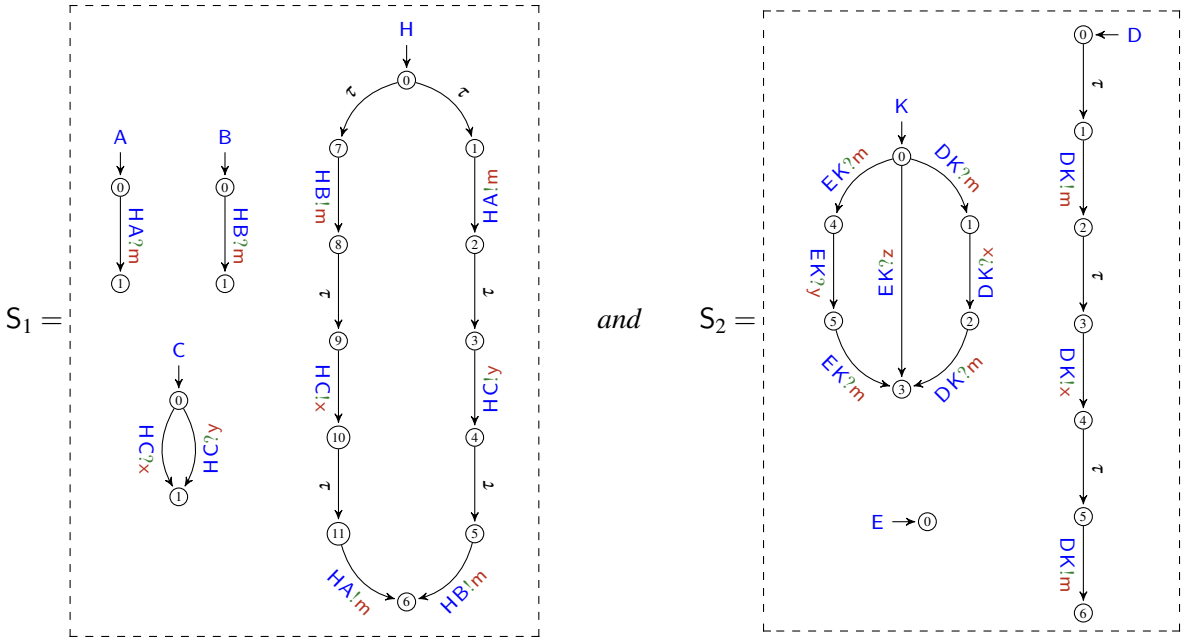
Notice that the initial states are asymmetric mixed and that H *and* K *are compatible. The gateways we obtain are*



The composed system $\mathsf{S}_1 {}^{\mathsf{H}\leftrightarrow\mathsf{K}} \mathsf{S}_2$ *deadlocks when* $\mathsf{gw}(\mathsf{S}_1(\mathsf{H}),\mathsf{K})$ *receives from* B *while* $\mathsf{gw}(\mathsf{S}_2(\mathsf{K}),\mathsf{H})$ *receives from* D *since both gateways reach an output state (respectively states* 10 *and* 7*).* ◇

The following examples show that, as asymmetric mixed states, !?-nondeterminism is problematic too. Let us first take two deadlock free systems.

Example 4.4. *The two systems*



are deadlock free.

The deadlock freedom of S_1 *follows from the fact that, from its initial configuration* $(0_\mathsf{A}, 0_\mathsf{B}, 0_\mathsf{C}, 0_\mathsf{H})$, S_1 *can only branch over the two* τ*-transitions of* H *reaching either of the following configurations*

$$(0_\mathsf{A}, 0_\mathsf{B}, 0_\mathsf{C}, 7_\mathsf{H}) \qquad \textit{or} \qquad (0_\mathsf{A}, 0_\mathsf{B}, 0_\mathsf{C}, 1_\mathsf{H})$$

From the former (resp. latter) configuration S_1 *can only reach configurations where* H *synchronises with* C *and then with* A *(resp.* B*). In either case* S_1 *reaches the terminal configuration* $(1_\mathsf{A}, 1_\mathsf{B}, 1_\mathsf{C}, 6_\mathsf{H})$.

Let us now have a look at S_2*. Firstly note that* E *cannot synchronise since it is already terminated; hence, the only possible transitions of* S_2 *must involve* K *and* D *only. We therefore have that*

$$\begin{aligned}(0_\mathsf{K}, 0_\mathsf{D}, 0_\mathsf{E}) &\xrightarrow{\tau} (0_\mathsf{K}, 1_\mathsf{D}, 0_\mathsf{E}) \xrightarrow{\mathsf{D}\to\mathsf{K}:\,\mathsf{m}} (1_\mathsf{K}, 2_\mathsf{D}, 0_\mathsf{E}) \xrightarrow{\tau} (1_\mathsf{K}, 3_\mathsf{D}, 0_\mathsf{E}) \xrightarrow{\mathsf{D}\to\mathsf{K}:\,\mathsf{x}} (2_\mathsf{K}, 4_\mathsf{D}, 0_\mathsf{E}) \\ &\xrightarrow{\tau} (2_\mathsf{K}, 5_\mathsf{D}, 0_\mathsf{E}) \xrightarrow{\mathsf{D}\to\mathsf{K}:\,\mathsf{m}} (3_\mathsf{K}, 6_\mathsf{D}, 0_\mathsf{E})\end{aligned}$$

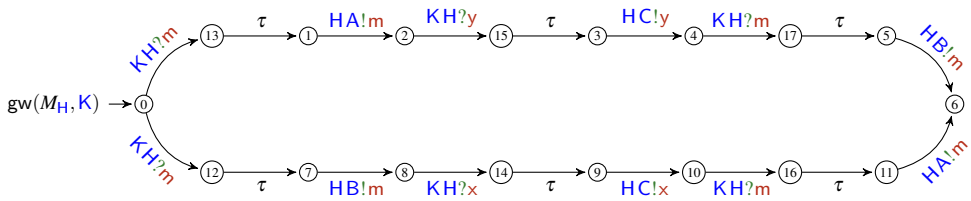
is the only possible execution from the initial configuration $(0_\mathsf{K}, 0_\mathsf{D}, 0_\mathsf{E})$ *of* S_2*, leading to the terminal configuration* $(3_\mathsf{K}, 6_\mathsf{D}, 0_\mathsf{E})$.

◇

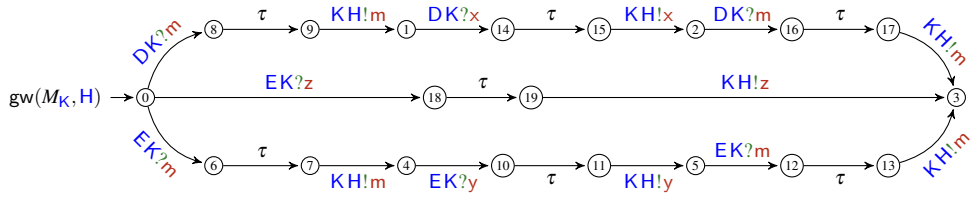
The next example shows that the compositions of the systems S_1 and S_2 in Example 4.4 can deadlock.

Example 4.5 (?!-determinism is necessary)**.** *The CFSMs* H *and* K *in Example 4.4 are compatible as seen*

in Example 3.6. Hence, we can build the composed system $\mathsf{S}_1 {}^{\mathsf{H}\leftrightarrow\mathsf{K}} \mathsf{S}_2$ *through the gateways*



and



Now, from the initial configuration $s_0 = (0_\mathsf{A}, 0_\mathsf{B}, 0_\mathsf{C}, 0_{\mathsf{gw}(M_\mathsf{H},\mathsf{K})}, 0_{\mathsf{gw}(M_\mathsf{K},\mathsf{H})}, 0_\mathsf{D}, 0_\mathsf{E})$ *of* $\mathsf{S}_1 {}^{\mathsf{H}\leftrightarrow\mathsf{K}} \mathsf{S}_2$ *we have the following run*

$$
\begin{aligned}
s_0 &\xrightarrow{\tau} (0_\mathsf{A}, 0_\mathsf{B}, 0_\mathsf{C}, 0_{\mathsf{gw}(M_\mathsf{H},\mathsf{K})}, 0_{\mathsf{gw}(M_\mathsf{K},\mathsf{H})}, 1_\mathsf{D}, 0_\mathsf{E}) \\
&\xrightarrow{\mathsf{D}\to\mathsf{K}:\,\mathsf{m}} \xrightarrow{\tau} (0_\mathsf{A}, 0_\mathsf{B}, 0_\mathsf{C}, 0_{\mathsf{gw}(M_\mathsf{H},\mathsf{K})}, 9_{\mathsf{gw}(M_\mathsf{K},\mathsf{H})}, 2_\mathsf{D}, 0_\mathsf{E}) \\
&\xrightarrow{\mathsf{K}\to\mathsf{H}:\,\mathsf{m}} \xrightarrow{\tau} (0_\mathsf{A}, 0_\mathsf{B}, 0_\mathsf{C}, 13_{\mathsf{gw}(M_\mathsf{H},\mathsf{K})}, 1_{\mathsf{gw}(M_\mathsf{K},\mathsf{H})}, 3_\mathsf{D}, 0_\mathsf{E}) \qquad (8) \\
&\xrightarrow{\mathsf{D}\to\mathsf{K}:\,\mathsf{x}} (0_\mathsf{A}, 0_\mathsf{B}, 0_\mathsf{C}, 13_{\mathsf{gw}(M_\mathsf{H},\mathsf{K})}, 14_{\mathsf{gw}(M_\mathsf{K},\mathsf{H})}, 4_\mathsf{D}, 0_\mathsf{E}) \\
&\xrightarrow{\tau}\xrightarrow{\tau}\xrightarrow{\tau} (0_\mathsf{A}, 0_\mathsf{B}, 0_\mathsf{C}, 1_{\mathsf{gw}(M_\mathsf{H},\mathsf{K})}, 15_{\mathsf{gw}(M_\mathsf{K},\mathsf{H})}, 5_\mathsf{D}, 0_\mathsf{E}) \\
&\xrightarrow{\mathsf{H}\to\mathsf{A}:\,\mathsf{m}} (1_\mathsf{A}, 0_\mathsf{B}, 0_\mathsf{C}, 2_{\mathsf{gw}(M_\mathsf{H},\mathsf{K})}, 15_{\mathsf{gw}(M_\mathsf{K},\mathsf{H})}, 5_\mathsf{D}, 0_\mathsf{E}) \qquad (9)
\end{aligned}
$$

where the τ*-transition of* D *enables the synchronisation of* $\mathsf{gw}(M_\mathsf{K},\mathsf{H})$ *and* D *with label* $\mathsf{D}\to\mathsf{K}:\mathsf{m}$ *that leads the gateway in state* 9 *after its* τ*-transition from state* 8*. Now, the two gateways can communicate and exchange message* m*. Due to* ?!*-nondeterminism of* S_1*, from state* 0 $\mathsf{gw}(M_\mathsf{H},\mathsf{K})$ *can move either to state* 12 *or to state* 13*. Fatally, transition* (8) *leads to a deadlock: after* $\mathsf{gw}(M_\mathsf{K},\mathsf{H})$ *and* D *synchronise to exchange message* x *the system goes into a configuration from where* $\mathsf{gw}(M_\mathsf{H},\mathsf{K})$ *forwards* m *to* A *and reaches the last configuration* (9)*. This is a deadlock for* $\mathsf{S}_1 {}^{\mathsf{H}\leftrightarrow\mathsf{K}} \mathsf{S}_2$*, since none of the CFSMs can do a* τ*-transitions, the only enabled output action is from* $\mathsf{gw}(M_\mathsf{K},\mathsf{H})$ *which tries to send message* x *to* $\mathsf{gw}(M_\mathsf{H},\mathsf{K})$*; however,* $\mathsf{gw}(M_\mathsf{H},\mathsf{K})$ *can only receive message* y *from* K *and hence these actions cannot synchronise.* ◇

5 Preserving Properties by Composition

Composition via gateways does not ensure the preservation of communication properties. We provide below sufficient conditions for this to happen. Recall that (H,K)-composability requires absence of asymmetric mixed states and ?!-determinism.

Theorem 5.1 (Deadlock freedom preservation)**.** *Let* S_1 *and* S_2 *be two* (H,K)*-composable and deadlock-free systems. Then the composed system* $\mathsf{S}_1 {}^{\mathsf{H}\leftrightarrow\mathsf{K}} \mathsf{S}_2$ *is deadlock-free.*

Proof sketch. The proof relies on the fact that the reachable configurations of $\mathsf{S}_1 {}^{\mathsf{H}\leftrightarrow\mathsf{K}} \mathsf{S}_2$ can be projected on reachable configurations of S_1 and S_2. This implies that a deadlock in $\mathsf{S}_1 {}^{\mathsf{H}\leftrightarrow\mathsf{K}} \mathsf{S}_2$ corresponds to a deadlock in S_1 or in S_2. See the appendix for the detailed proof. □

Example 5.2. *We can infer deadlock-freedom of the system* $\mathsf{S} = \mathsf{S}_1 {}^{\mathsf{H}\leftrightarrow\mathsf{K}} \mathsf{S}_2$ *of Example 3.4 by the result above, since* S_1 *and* S_1 *are* (H,K)*-composable and deadlock-free.*

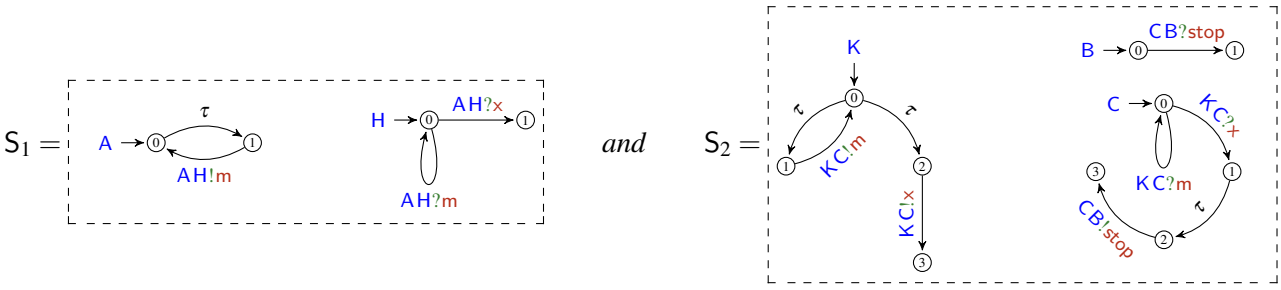
Somehow surprisingly, in the symmetric case preservation of deadlock freedom requires stricter conditions on gateways than in the asymmetric case. In fact, in the asymmetric case, deadlock freedom preservation requires only absence of asymmetric mixed states and ?!-determinism while the symmetric case requires the (stronger) condition of *sequentiality*.

Definition 5.3 (Sequential CFSM)**.** *A CFSM is* sequential *if each of its states has at most one outgoing transition. A participant* A *of a system* S *is sequential if* $\mathsf{S}(\mathsf{A})$ *is so.*

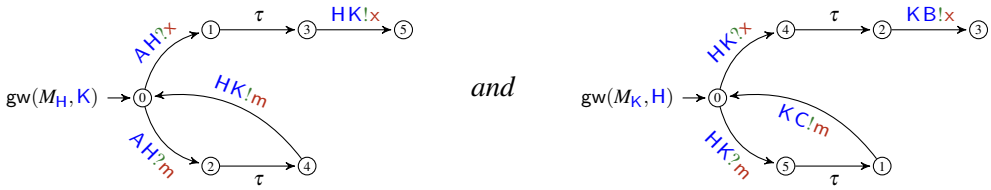
As we will see (cf. Theorem 5.5), sequentiality is necessary to preserve lock-freedom also in the asymmetric case. We note that sequentiality implies absence of asymmetric mixed states and ?!-determinism, while the converse does not hold.

As mentioned before, the property of lock freedom is not preserved in general by composition, as shown by the following example.

Example 5.4 (Composability does not preserve lock-freedom)**.** *Take the communicating systems*



Note that both S_1 *and* S_2 *are lock-free and that* H *and* K *are compatible. The gateways are*



Hence, the composed system $\mathsf{S}_1 {}^{\mathsf{H}\leftrightarrow\mathsf{K}} \mathsf{S}_2$ *is non lock-free because e.g. the configuration*

$$s = (0_{\mathsf{A}}, 0_{\mathsf{gw}(M_{\mathsf{H}},\mathsf{K})}, 0_{\mathsf{gw}(M_{\mathsf{K}},\mathsf{H})}, 0_{\mathsf{B}}, 0_{\mathsf{C}})$$

is a lock for B*, since the only outgoing transition from* 0_{B} *could be fired only in case the transition* $\mathsf{CB!stop}$ *is enabled. However, this is impossible since* $\mathsf{gw}(M_{\mathsf{H}},\mathsf{K})$ *forwards only message* m*; hence, the run (which does not involve* B*)*

$$s \xrightarrow{\tau} \xrightarrow{\mathsf{A}\to\mathsf{H}:\,\mathsf{m}} (0_{\mathsf{A}}, 2_{\mathsf{gw}(M_{\mathsf{H}},\mathsf{K})}, 0_{\mathsf{gw}(M_{\mathsf{K}},\mathsf{H})}, 0_{\mathsf{B}}, 0_{\mathsf{C}}) \xrightarrow{\tau} \xrightarrow{\mathsf{A}\to\mathsf{H}:\,\mathsf{m}} s \cdots$$

is perpetually executed. ◇

We show that the problem of Example 5.4 cannot happen in case we restrict to sequential gateways, as done for deadlock freedom in the symmetric case (cf. [5]). As usual, $f\,|_X$ denotes the restriction of a function f on a subset X of its domain.

Theorem 5.5 (Lock-freedom preservation)**.** *Let* S_1 *and* S_2 *be two* (H,K)*-composable and lock-free systems with* H *and* K *sequential. Then the composed system* $\mathsf{S}_1 {}^{\mathsf{H}\leftrightarrow\mathsf{K}} \mathsf{S}_2$ *is lock-free.*

Proof sketch. The proof goes as the one of Theorem 5.1 noticing that we have to reconstruct "backward" the sequence of interactions. This exploits sequentiality and lock-freedom of S_1 and S_2 in order to guarantee the reconstruction when we "cross" the two composed systems through the gateways. □

We turn now our attention to strong lock-freedom. In this case, as for deadlock freedom, (H,K)-composability suffices for preservation by composition; we shall see that this is not the case for lock freedom preservation.

Theorem 5.6 (Strong lock freedom preservation)**.** *Let* S_1 *and* S_2 *be two* (H,K)*-composable and strongly lock free systems. Then the composed system* $\mathsf{S}_1 {}^{\mathsf{H}\leftrightarrow\mathsf{K}} \mathsf{S}_2$ *is strongly lock free.*

Proof sketch. The proof is similar to the one of Theorem 5.5 but for the use of strong lock freedom of S_1 and S_2 instead of their deadlock freedom. □

6 Conclusions and Future Work

We introduce an asymmetric synchronous semantics of communicating systems which breaks the symmetry between senders and receivers. In fact, our semantics decouples communication from choice resolution as in standard semantics of communicating systems (and other models). We then adapted the gateway composition mechanism defined in [2, 3] to our asymmetric semantics and gave conditions for the preservation of some communication properties under this notion of composition.

An approach related to ours is the framework of [8, 9] based on *contract automata* where transitions express "requests" and "offers" among participants. The composition mechanism is based on "trimming" a product of contract automata according to relevant *agreement* properties. This yields controllers that preserve deadlocks. Contract automata do not consider asymmetric synchronous semantics. Our composition mechanism does not introduce orchestrators which, under some conditions, can be avoided also for contract automata [8, 9].

Modular approaches to the development of concurrent systems can be exploited even for systems designed using formalisms intrinsically dealing with *closed* systems. Indeed, given two systems, any two components – one per system – exhibiting *compatible* behaviours can be replaced by two coupled forwarders (gateways) connecting the systems, as investigated initially in [2, 3] for an asynchronous interaction model. The investigation on the composition-by-gateways technique was shifted in [5] towards synchronous symmetric interactions. In the present paper we pushed a step forward such an investigation, by considering *asymmetric* synchronous interactions. Interestingly, deadlock freedom preservation in the synchronous asymmetric case we consider does not require sequentiality of gateways, like in the asynchronous case, and differently from the synchronous symmetric case. Notably, sequentiality is needed here for lock-freedom preservation, but not for strong-lock freedom preservation.

While the path of investigation above is quite homogeneous, the different analyses present some methodological differences. For instance, [5] considers also another form of composition, where one single gateway (interacting with both the composed systems) is used. On the other side, [5] focused only

on deadlocks, disregarding other properties we consider. A first item of future research consist in filling the bits missing due to the mismatches above.

A more challenging direction for future work is looking for refined composition mechanisms in order to get preservation of relevant properties under weaker conditions.

References

[1] F. Barbanera and U. de'Liguoro. Sub-behaviour relations for session-based client/server systems. *MSCS*, 25(6):1339–1381, 2015.

[2] 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.

[3] F. Barbanera, U. de'Liguoro, and R. Hennicker. Connecting open systems of communicating finite state machines. *JLAMP*, 109, 2019.

[4] F. Barbanera, M. Dezani-Ciancaglini, I. Lanese, and E. Tuosto. Composition and decomposition of multiparty sessions. *JLAMP*, 2020. Submitted.

[5] F. Barbanera, I. Lanese, and E. Tuosto. Composing communicating systems, synchronously. In T. Margaria and B. Steffen, editors, *ISoLA 2020*, volume 12476 of *LNCS*, pages 39–59. Springer, 2020.

[6] M. Bartoletti, T. Cimoli, and R. Zunino. Compliance in behavioural contracts: A brief survey. In C. Bodei, G. L. Ferrari, and C. Priami, editors, *Programming Languages with Applications to Biology and Security - Essays Dedicated to Pierpaolo Degano on the Occasion of His 65th Birthday*, volume 9465 of *Lecture Notes in Computer Science*, pages 103–121. Springer, 2015.

[7] M. Bartoletti, A. Scalas, and R. Zunino. A semantic deconstruction of session types. In P. Baldan and D. Gorla, editors, *CONCUR 2014 - Concurrency Theory - 25th International Conference, CONCUR 2014, Rome, Italy, September 2-5, 2014. Proceedings*, volume 8704 of *Lecture Notes in Computer Science*, pages 402–418. Springer, 2014.

[8] D. Basile, P. Degano, G. L. Ferrari, and E. Tuosto. Playing with our CAT and communication-centric applications. In E. Albert and I. Lanese, editors, *FORTE*, volume 9688 of *LNCS*, pages 62–73. Springer, 2016.

[9] D. Basile, M. H. ter Beek, and R. Pugliese. Synthesis of orchestrations and choreographies: Bridging the gap between supervisory control and coordination of services. *LMCS*, 16(2), 2020.

[10] G. Bernardi and M. Hennessy. Modelling session types using contracts. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, SAC '12, pages 1941–1946, New York, NY, USA, 2012. ACM.

[11] D. Brand and P. Zafiropulo. On communicating finite-state machines. *J. ACM*, 30(2):323–342, 1983.

[12] G. Cécé and A. Finkel. Verification of programs with half-duplex communication. *I&C*, 202(2):166–190, 2005.

[13] K. Honda, V. T. Vasconcelos, and M. Kubo. Language primitives and type disciplines for structured communication-based programming. In *ESOP*, volume 1381 of *LNCS*, pages 22–138. Springer, 1998.

[14] H. Hüttel et al. Foundations of session types and behavioural contracts. *ACM Comput. Surv.*, 49(1):3:1–3:36, 2016.

[15] J. Lange, E. Tuosto, and N. Yoshida. From communicating machines to graphical choreographies. In *POPL*, pages 221–232. ACM, 2015.

[16] R. Milner. *A Calculus of Communicating Systems*, volume 92 of *LNCS*. Springer, Berlin, 1980.

[17] L. Padovani. Contract-based discovery of web services modulo simple orchestrators. *Theoretical Computer Science*, 411:3328–3347, 2010.

[18] E. Tuosto and R. Guanciale. Semantics of global view of choreographies. *JLAMP*, 95:17–40, 2018.

A Proofs for Section 3 (Composition via Gateways)

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_1 and S_2 while avoiding, for the gateways, to take the fresh states introduced by the gateway construction. Indeed, we shall prove in Proposition B.3 that for each $s \in \mathscr{R}([\![\mathsf{S}]\!])$, we have ${}_{\mathsf{H}}\rfloor s \in \mathscr{R}([\![\mathsf{S}_1]\!])$ and $s\lfloor_{\mathsf{K}} \in \mathscr{R}([\![\mathsf{S}_2]\!])$.

Definition A.1 (Configuration projections)**.** *Let s be a configuration of a composed system $\mathsf{S}_1{}^{\mathsf{H}\leftrightarrow\mathsf{K}}\mathsf{S}_2$ and $M = \mathsf{gw}(\mathsf{S}_1(\mathsf{H}),\mathsf{K})$. The* left-projection of s on S_1 *is the map ${}_{\mathsf{H}}\rfloor s$ defined on* $\mathrm{dom}(\mathsf{S}_1)$ *by*

$$ {}_{\mathsf{H}}\rfloor s : \mathsf{A} \mapsto \begin{cases} q, & \text{if } s(\mathsf{A}) \notin \mathsf{S}_1(\mathsf{A}) \text{ and there is } q \xrightarrow{\mathsf{K}\,\mathsf{H}?\mathsf{m}} s(\mathsf{A}) \in M \text{ or } q \xrightarrow{\mathsf{K}\,\mathsf{H}?\mathsf{m}} \xrightarrow{\tau} s(\mathsf{A}) \in M \\ r, & \text{if } s(\mathsf{A}) \notin \mathsf{S}_1(\mathsf{A}) \text{ and there is } s(\mathsf{A}) \xrightarrow{\mathsf{H}\,\mathsf{K}!\mathsf{m}} r \in M \text{ or } s(\mathsf{A}) \xrightarrow{\tau} \xrightarrow{\mathsf{H}\,\mathsf{K}!\mathsf{m}} r \in M \\ s(\mathsf{A}), & \text{otherwise} \end{cases} $$

The definition of right-projection $s\lfloor_{\mathsf{K}}$ *is analogous.*

Proposition A.2. *Left- and right-projections are well-defined.*

Proof. It is enough to show that ${}_{\mathsf{H}}\rfloor\cdot$ and $\cdot\lfloor_{\mathsf{K}}$ uniquely assign a state to fresh states because on non-fresh stats both functions act as the identify map. This follows since, by construction, each state introduced by our gateway construction has unique successor and predecessor. □

Intuitively, only H is aware of an input from K when H is in the internal state reached after an input from K; hence to have a coherent configuration we take the state of H before the input. If instead H is in an internal state corresponding to an output to K, then 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 $s\lfloor_{\mathsf{K}}$.)

Example A.3. *Let $s = (2_{\mathsf{A}}, 0_{\mathsf{B}}, 1_{\mathsf{H}}, 5_{\mathsf{K}}, 0_{\mathsf{C}}, 0_{\mathsf{D}}, 0_{\mathsf{E}})$; and $\mathsf{S} = \mathsf{S}_1{}^{\mathsf{H}\leftrightarrow\mathsf{K}}\mathsf{S}_2$ be the system of Example 3.4. Then, $s \in \mathscr{R}([\![\mathsf{S}]\!])$, namely s is reachable in S. In fact*

$$ \begin{aligned} s_0 = (0_{\mathsf{A}}, 0_{\mathsf{B}}, 0_{\mathsf{H}}, 0_{\mathsf{K}}, 0_{\mathsf{C}}, 0_{\mathsf{D}}, 0_{\mathsf{E}}) &\xrightarrow{\tau} (1_{\mathsf{A}}, 0_{\mathsf{B}}, 0_{\mathsf{H}}, 0_{\mathsf{K}}, 0_{\mathsf{C}}, 0_{\mathsf{D}}, 0_{\mathsf{E}}) \\ &\xrightarrow{\mathsf{A}\to\mathsf{H}:\,\mathsf{m}} (2_{\mathsf{A}}, 0_{\mathsf{B}}, 2_{\mathsf{H}}, 0_{\mathsf{K}}, 0_{\mathsf{C}}, 0_{\mathsf{D}}, 0_{\mathsf{E}}) \\ &\xrightarrow{\tau} (2_{\mathsf{A}}, 0_{\mathsf{B}}, 4_{\mathsf{H}}, 0_{\mathsf{K}}, 0_{\mathsf{C}}, 0_{\mathsf{D}}, 0_{\mathsf{E}}) \\ &\xrightarrow{\mathsf{H}\to\mathsf{K}:\,\mathsf{m}} (2_{\mathsf{A}}, 0_{\mathsf{B}}, 1_{\mathsf{H}}, 5_{\mathsf{K}}, 0_{\mathsf{C}}, 0_{\mathsf{D}}, 0_{\mathsf{E}}) \end{aligned} $$

The projections of s on, respectively, S_1 and S_2 are ${}_{\mathsf{H}}\rfloor s = (2_{\mathsf{A}}, 0_{\mathsf{B}}, 1_{\mathsf{H}})$ and $s\lfloor_{\mathsf{K}} = (0_{\mathsf{K}}, 0_{\mathsf{C}}, 0_{\mathsf{D}}, 0_{\mathsf{E}})$. ◇

B Proofs for Section 4 (Composition Related Issues)

Let M be an H-local CFSM and $\mathsf{K} \in \mathscr{P} \setminus \{\mathsf{H}\}$ be a participant not occurring in M. Function nof maps the states of $\mathsf{gw}(M,\mathsf{K})$ to the states of M as follows:

$$ \mathsf{nof}_{\mathsf{H},\mathsf{K}}(M,q) = \begin{cases} p, & \text{if } p \xrightarrow{\lambda} q \in \mathsf{gw}(M,\mathsf{K}) \text{ and } \lambda \text{ input label with } \mathsf{K} \notin \mathsf{ptp}(\lambda) \\ p, & \text{if } p \xrightarrow{\lambda} p' \xrightarrow{\tau} q \in \mathsf{gw}(M,\mathsf{K}) \text{ and } \lambda \text{ input label with } \mathsf{K} \in \mathsf{ptp}(\lambda) \\ r, & \text{if } q \xrightarrow{\lambda} r \in \mathsf{gw}(M,\mathsf{K}) \text{ and } \lambda \text{ output label with } \mathsf{K} \notin \mathsf{ptp}(\lambda) \\ r, & \text{if } q \xrightarrow{\tau} q' \xrightarrow{\lambda} r \in \mathsf{gw}(M,\mathsf{K}) \text{ and } \lambda \text{ output label with } \mathsf{K} \in \mathsf{ptp}(\lambda) \\ q, & \text{if } q \text{ is a state of } M \end{cases} $$

Note that the first four clause imply that q is a fresh state of $\mathsf{gw}(M,\mathsf{K})$.

Lemma B.1. *Function* nof *is well-defined.*

Proof. Let M be an H-local CFSM and $\mathsf{K} \in \mathscr{P} \setminus \{\mathsf{H}\}$. We have to check only internal states as the restriction of nof to the states of M is the identity by definition. If q is an internal state of $\mathsf{gw}(M,\mathsf{K})$, by definition of $\mathsf{gw}(M,\mathsf{K})$, there is a unique q' such that either $q' \xrightarrow{\mathsf{A}\mathsf{H}?\mathsf{m}} q \in \mathsf{gw}(M,\mathsf{K})$ or $q \xrightarrow{\mathsf{H}\mathsf{A}!\mathsf{m}} q' \in \mathsf{gw}(M,\mathsf{K})$. □

Example B.2. *For* $\mathsf{S}_1 {}^{\mathsf{H}\leftrightarrow\mathsf{K}} \mathsf{S}_2$ *of Example 3.4,* $\mathsf{nof}_{\mathsf{H},\mathsf{K}}(\mathsf{S}_1(\mathsf{H}),0) = 0$ *and* $\mathsf{nof}_{\mathsf{K},\mathsf{H}}(\mathsf{S}_2(\mathsf{K}),2) = 0$. ◇

Function nof is similar to configuration projection when considering CFSMs in isolation, with a main difference: when e.g., $\mathsf{gw}(M,\mathsf{K})$ in state q receives from a partner in its own system going to some fresh state p' with a τ-transition to p'', nof maps both p' and p'' to p (unlike configuration projection $_{\mathsf{H}}\rfloor_-$ which maps q to the successor of p''). This represents the fact that the other system, and K in particular, are oblivious of the transition. In fact, function nof is designed to establish a correspondence with the other system as shown by the next proposition.

Proposition B.3. *Let* $\mathsf{S} = \mathsf{S}_1 {}^{\mathsf{H}\leftrightarrow\mathsf{K}} \mathsf{S}_2$ *be the composition of two* (H,K)*-composable systems* S_1 *and* S_2*. If* $s \in \mathscr{R}([\![\mathsf{S}]\!])$ *then* $_{\mathsf{H}}\rfloor s \in \mathscr{R}([\![\mathsf{S}_1]\!])$*,* $s\lfloor_{\mathsf{K}} \in \mathscr{R}([\![\mathsf{S}_2]\!])$*, and* $\mathsf{nof}_{\mathsf{H},\mathsf{K}}(\mathsf{S}_1(\mathsf{H}),s(\mathsf{H})) \asymp \mathsf{nof}_{\mathsf{K},\mathsf{H}}(\mathsf{S}_2(\mathsf{K}),s(\mathsf{K}))$.

Proof. Let s_0 be the initial configuration of S and

$$s_0 \xrightarrow{\lambda_1} s_1 \cdots s_{n-1} \xrightarrow{\lambda_n} s_n = s \tag{10}$$

a run reaching s from s_0. We proceed by induction on n.

If $n = 0$ the thesis is immediate by observing that

- $_{\mathsf{H}}\rfloor s \in \mathscr{R}([\![\mathsf{S}_1]\!])$ and $s\lfloor_{\mathsf{K}} \in \mathscr{R}([\![\mathsf{S}_2]\!])$ because left- and right-projections are the initial configurations of S_1 and S_2 respectively, and
- $\mathsf{nof}_{\mathsf{H},\mathsf{K}}(\mathsf{S}_1(\mathsf{H}),s(\mathsf{H}))$ and $\mathsf{nof}_{\mathsf{K},\mathsf{H}}(\mathsf{S}_2(\mathsf{K}),s(\mathsf{K}))$ are the initial states of $\mathsf{S}_1(\mathsf{H})$ and $\mathsf{S}_2(\mathsf{K})$ respectively which are compatible by hypothesis.

Let $n > 0$ and assume that the statement holds for all configurations reachable from s_0 in less the n transitions. We have that either none of H and K are involved in λ_n or that at least one of them is. In the former case, without loss of generality, assume that the interacting participants, say A and B are both in S_1. Then, by construction (cf. Definition A.1), $s\lfloor_{\mathsf{K}} = s_{n-1}\lfloor_{\mathsf{K}}$ and by inductive hypothesis $s_{n-1}\lfloor_{\mathsf{K}} \in \mathscr{R}([\![\mathsf{S}_2]\!])$. Moreover, $_{\mathsf{H}}\rfloor s$ equals $_{\mathsf{H}}\rfloor s_{n-1}$ but for the local states of A and B; hence $_{\mathsf{H}}\rfloor s_{n-1} \in \mathscr{R}([\![\mathsf{S}_1]\!])$ by the semantics of communicating systems (cf. Definition 2.3). Also,

$$\mathsf{nof}_{\mathsf{H},\mathsf{K}}(\mathsf{S}_1(\mathsf{H}),s(\mathsf{H})) = (_{\mathsf{H}}\rfloor s)(\mathsf{H}) = (_{\mathsf{H}}\rfloor s_{n-1})(\mathsf{H}) \asymp (s_{n-1}\lfloor_{\mathsf{K}})(\mathsf{K}) = (s\lfloor_{\mathsf{K}})(\mathsf{K}) = \mathsf{nof}_{\mathsf{K},\mathsf{H}}(\mathsf{S}_2(\mathsf{K}),s(\mathsf{K}))$$

because the equalities above hold by the definition of asymmetric synchronisation (cf. Definition 2.3) and the compatibility relation holds by the inductive hypothesis. So, let us assume that at least one between H and K is involved in the last transition reaching s and proceed by case analysis on λ_n.

$\boxed{\lambda_n = \tau}$ We consider only the case where H is involved since the case where K is involved is symmetric. By our gateway construction (cf. Definition 3.1) only one of the following two cases are possible for the transitions in $\mathsf{gw}(\mathsf{S}_1(\mathsf{H}),\mathsf{K})$:

$$p \xrightarrow{\mathsf{K}\mathsf{H}?\mathsf{m}} s_{n-1}(\mathsf{H}) \xrightarrow{\tau} q \xrightarrow{\mathsf{H}\mathsf{A}!\mathsf{m}} r \tag{11}$$

$$q \xrightarrow{\mathsf{A}\mathsf{H}?\mathsf{m}} s_{n-1}(\mathsf{H}) \xrightarrow{\tau} p \xrightarrow{\mathsf{H}\mathsf{K}!\mathsf{m}} r \tag{12}$$

for some participant A of S_1 and states p, q, r of $\mathsf{gw}(\mathsf{S}_1(\mathsf{H}),\mathsf{K})$. Cases (11) and (12) respectively correspond to have the transitions

$$p \xrightarrow{\tau} q \xrightarrow{\mathsf{HA!m}} r \qquad \text{and} \qquad q \xrightarrow{\mathsf{AH?m}} r$$

in the machine $\mathsf{S}_1(\mathsf{H})$.

In case (11) the last transition in the run (10) must therefore be preceded by a transition, say the i-th one, where the machines $\mathsf{gw}(\mathsf{S}_1(\mathsf{H}),\mathsf{K})$ and $\mathsf{gw}(\mathsf{S}_2(\mathsf{K}),\mathsf{H})$ have exchanged message m. Hence, we have that $\mathsf{gw}(\mathsf{S}_2(\mathsf{K}),\mathsf{H})$ has a transition $s_i(\mathsf{K}) \xrightarrow{\mathsf{KH!m}} s_{i+1}(\mathsf{K})$ with $s_{n-1}(\mathsf{H}) \asymp s_{i+1}(\mathsf{K})$ since (by inductive hypothesis) $s_i(\mathsf{H}) = p = \mathsf{nof}_{\mathsf{H},\mathsf{K}}(\mathsf{S}_1(\mathsf{H}),s_i(\mathsf{H})) \asymp \mathsf{nof}_{\mathsf{K},\mathsf{H}}(\mathsf{S}_2(\mathsf{K}),s_i(\mathsf{K}))$ and both gateways are ?!-deterministic. We now observe that either $s(\mathsf{K}) = s_{i+1}(\mathsf{K})$ or it is a fresh state that $\mathsf{gw}(\mathsf{S}_2(\mathsf{K}),\mathsf{H})$ reaches after having received a message from a participant of S_2 (possibly followed by a τ transition) between the i-th and the last interactions in (10). We have that ${}_{\mathsf{H}}\lfloor s = {}_{\mathsf{H}}\lfloor s_{n-1}[\mathsf{H} \mapsto p]$, hence ${}_{\mathsf{H}}\lfloor s \in \mathscr{R}(\llbracket \mathsf{S}_1 \rrbracket)$ by inductive hypothesis; also, $s\lfloor_{\mathsf{K}} \in \mathscr{R}(\llbracket \mathsf{S}_2 \rrbracket)$ by the inductive hypothesis since $s\lfloor_{\mathsf{K}} = s_{n-1}\lfloor_{\mathsf{K}}$ because K is not involved in λ_n. Finally, in both cases the last part of the thesis immediately follows by observing that $\mathsf{nof}_{\mathsf{K},\mathsf{H}}(\mathsf{S}_2(\mathsf{K}),s(\mathsf{K})) = \mathsf{nof}_{\mathsf{K},\mathsf{H}}(\mathsf{S}_2(\mathsf{K}),s_i(\mathsf{K}))$ by definition and $\mathsf{nof}_{\mathsf{H},\mathsf{K}}(\mathsf{S}_1(\mathsf{H}),s(\mathsf{H})) = p$.

In case (12), firstly note that by construction $\mathsf{A} \neq \mathsf{H}$ (cf. Definition 2.3). Then $\mathsf{gw}(\mathsf{S}_1(\mathsf{H}),\mathsf{K})$ has the transition $s_{n-1}(\mathsf{H}) \xrightarrow{\mathsf{AH?m}} s(\mathsf{H})$ which corresponds to an input transition $s_{n-1}(\mathsf{H}) \xrightarrow{\mathsf{AH?m}} r$ where $r = {}_{\mathsf{H}}\lfloor s(\mathsf{H}) = \mathsf{nof}_{\mathsf{H},\mathsf{K}}(\mathsf{S}_1(\mathsf{H}),s_{n-1}(\mathsf{H})) = \mathsf{nof}_{\mathsf{H},\mathsf{K}}(\mathsf{S}_1(\mathsf{H}),s(\mathsf{H}))$ in $\mathsf{S}_1(\mathsf{H})$. Hence, the thesis follows since $s\lfloor_{\mathsf{K}} = s_{n-1}\lfloor_{\mathsf{K}}$ because K is not involved in λ_n and $\mathsf{nof}_{\mathsf{H},\mathsf{K}}(\mathsf{S}_1(\mathsf{H}),s(\mathsf{H})) \asymp \mathsf{nof}_{\mathsf{K},\mathsf{H}}(\mathsf{S}_2(\mathsf{K}),s(\mathsf{K})) = \mathsf{nof}_{\mathsf{K},\mathsf{H}}(\mathsf{S}_2(\mathsf{K}),s_{n-1}(\mathsf{K}))$ by inductive hypothesis.

$\boxed{\lambda_n = \mathsf{H} \to \mathsf{X}\colon \mathsf{m}}$ By construction, either $\mathsf{X} = \mathsf{K}$ or $\mathsf{X} \neq \mathsf{H}$ is a participant of S_1. In the former case, $\mathsf{gw}(\mathsf{S}_1(\mathsf{H}),\mathsf{K})$ has the transition $s_{n-1}(\mathsf{H}) \xrightarrow{\mathsf{HK!m}} s(\mathsf{H})$ which corresponds to an input transition while $\mathsf{gw}(\mathsf{S}_2(\mathsf{K}),\mathsf{H})$ has the transition $s_{n-1}(\mathsf{K}) \xrightarrow{\mathsf{HK?m}} s(\mathsf{K})$ which corresponds to a sequence of transitions $p \xrightarrow{\tau} \xrightarrow{\mathsf{KB!m}} s(\mathsf{K})$ in $\mathsf{S}_2(\mathsf{K})$ for some participant $\mathsf{B} \neq \mathsf{K}$ in S_2. Then the thesis follows by the fact that ${}_{\mathsf{H}}\lfloor s = {}_{\mathsf{H}}\lfloor s_{n-1}[\mathsf{H} \mapsto s(\mathsf{H})]$ and $s\lfloor_{\mathsf{K}} = s_{n-1}\lfloor_{\mathsf{K}}[\mathsf{K} \mapsto p]$ by construction (cf. Definition A.1) and that, by inductive hypothesis, $\mathsf{nof}_{\mathsf{H},\mathsf{K}}(\mathsf{S}_1(\mathsf{H}),s_{n-1}(\mathsf{H})) = q \asymp s(\mathsf{K}) = \mathsf{nof}_{\mathsf{K},\mathsf{H}}(\mathsf{S}_2(\mathsf{K}),s_{n-1}(\mathsf{K}))$ and therefore, by ?!-determinism and the compatibility relation $s(\mathsf{H}) \asymp s(\mathsf{K})$.

Suppose now that X is a participant of S_1; note that by construction $\mathsf{X} \neq \mathsf{H}$ (cf. Definition 2.3). Since $s\lfloor_{\mathsf{K}} = s_{n-1}\lfloor_{\mathsf{K}}$, the inductive hypothesis immediately entails that $s\lfloor_{\mathsf{K}} \in \mathscr{R}(\llbracket \mathsf{S}_2 \rrbracket)$.

We first show the reachability of left- and right-projections. The transition $s_{n-1}(\mathsf{H}) \xrightarrow{\mathsf{HX!m}} s(\mathsf{H})$ is in $\mathsf{gw}(\mathsf{S}_1(\mathsf{H}),\mathsf{K})$ by construction and it corresponds to a pair of transitions $p \xrightarrow{\tau} s_{n-1}(\mathsf{H}) \xrightarrow{\mathsf{HX!m}} s(\mathsf{H})$ in $\mathsf{S}_1(\mathsf{H})$. We have that ${}_{\mathsf{H}}\lfloor s_{n-1} \in \mathscr{R}(\llbracket \mathsf{S}_1 \rrbracket)$ (by inductive hypothesis) and since ${}_{\mathsf{H}}\lfloor s_{n-1}(\mathsf{H}) = p$ (by Definition A.1) we have ${}_{\mathsf{H}}\lfloor s_{n-1} \xrightarrow{\tau} \xrightarrow{\mathsf{H}\to\mathsf{X}\colon \mathsf{m}} {}_{\mathsf{H}}\lfloor s$ (by Definition 2.3).

We now show the compatibility condition. The last transition in the run (10) must be preceded by a transition, say the i-th one, where the machines $\mathsf{gw}(\mathsf{S}_1(\mathsf{H}),\mathsf{K})$ and $\mathsf{gw}(\mathsf{S}_2(\mathsf{K}),\mathsf{H})$ have exchanged message m. Hence, we have that $\mathsf{gw}(\mathsf{S}_2(\mathsf{K}),\mathsf{H})$ has a transition $s_i(\mathsf{K}) \xrightarrow{\mathsf{KH!m}} s_{i+1}(\mathsf{K})$ with

$$\mathsf{nof}_{\mathsf{H},\mathsf{K}}(\mathsf{S}_1(\mathsf{H}),s_{n-1}(\mathsf{H})) = s(\mathsf{H}) = \mathsf{nof}_{\mathsf{H},\mathsf{K}}(\mathsf{S}_1(\mathsf{H}),s(\mathsf{H})) \tag{13}$$

which hold by definition of $\mathsf{nof}(\_,\_)$. We now observe that either $s(\mathsf{K}) = s_{i+1}(\mathsf{K})$ or it is a fresh state that $\mathsf{gw}(\mathsf{S}_2(\mathsf{K}),\mathsf{H})$ reaches after having received a message from a participant of S_2 (possibly

followed by a τ transition) between the i-th and the last interactions in (10). In both cases the $\mathsf{nof}_{\mathsf{K},\mathsf{H}}(\mathsf{S}_2(\mathsf{K}), s(\mathsf{K})) = \mathsf{nof}_{\mathsf{K},\mathsf{H}}(\mathsf{S}_2(\mathsf{K}), s_{n-1}(\mathsf{K}))$ and, by inductive hypothesis,

$$\mathsf{nof}_{\mathsf{K},\mathsf{H}}(\mathsf{S}_2(\mathsf{K}), s_{n-1}(\mathsf{K})) \asymp \mathsf{nof}_{\mathsf{H},\mathsf{K}}(\mathsf{S}_1(\mathsf{H}), s_{n-1}(\mathsf{H}))$$

hence, by equalities (13), $\mathsf{nof}_{\mathsf{H},\mathsf{K}}(\mathsf{S}_1(\mathsf{H}), s(\mathsf{H})) \asymp \mathsf{nof}_{\mathsf{K},\mathsf{H}}(\mathsf{S}_2(\mathsf{K}), s(\mathsf{K}))$.

$\boxed{\lambda_n = \mathsf{X} \to \mathsf{H} : \mathsf{m}}$ The case $\mathsf{X} = \mathsf{K}$ is symmetric to the previous case with $\lambda_n = \mathsf{H} \to \mathsf{K} : \mathsf{m}$ and therefore omitted. So, assume that X is a participant of S_1; note that by construction $\mathsf{X} \neq \mathsf{H}$ (cf. Definition 2.3). Then $\mathsf{gw}(\mathsf{S}_1(\mathsf{H}), \mathsf{K})$ has the transition $s_{n-1}(\mathsf{H}) \xrightarrow{\mathsf{X}\,\mathsf{H}?\mathsf{m}} s(\mathsf{H})$ which corresponds to an input transition $s_{n-1}(\mathsf{H}) \xrightarrow{\mathsf{X}\,\mathsf{H}?\mathsf{m}} r$. We have that ${}_{\mathsf{H}}\lfloor s_{n-1} \in RS[[\![\mathsf{S}_1]\!]]$ by inductive hypothesis, and so is ${}_{\mathsf{H}}\lfloor s_{n-1} \xrightarrow{\mathsf{X} \to \mathsf{H} : \mathsf{m}} {}_{\mathsf{H}}\lfloor s$ since, by definition of left-projection, ${}_{\mathsf{H}}\lfloor s(\mathsf{H}) = r$. The reachability of the right-projection of s immediately follows by inductive hypothesis, since $s\lfloor_{\mathsf{K}} = s_{n-1}\lfloor_{\mathsf{K}}$.
We now show the compatibility condition. By definition, we have that $\mathsf{nof}_{\mathsf{H},\mathsf{K}}(\mathsf{S}_1(\mathsf{H}), s(\mathsf{H})) = \mathsf{nof}_{\mathsf{H},\mathsf{K}}(\mathsf{S}_1(\mathsf{H}), s_{n-1}(\mathsf{H}))$. Moreover, we necessarily have that $s(\mathsf{K}) = s_{n-1}(\mathsf{K})$. Hence by inductive hypothesis,

$$\mathsf{nof}_{\mathsf{H},\mathsf{K}}(\mathsf{S}_1(\mathsf{H}), s(\mathsf{H})) = \mathsf{nof}_{\mathsf{H},\mathsf{K}}(\mathsf{S}_1(\mathsf{H}), s_{n-1}(\mathsf{H})) \asymp \mathsf{nof}_{\mathsf{K},\mathsf{H}}(\mathsf{S}_2(\mathsf{K}), s_{n-1}(\mathsf{K})) = \mathsf{nof}_{\mathsf{K},\mathsf{H}}(\mathsf{S}_2(\mathsf{K}), s(\mathsf{K}))$$

The cases $\lambda_n = \mathsf{X} \to \mathsf{K} : \mathsf{m}$ and $\lambda_n = \mathsf{K} \to \mathsf{X} : \mathsf{m}$ are similar to the last two cases above. □

C Proofs for Section 5 (Preserving Properties by Composition)

Theorem 5.1 (Deadlock freedom preservation)**.** *Let* S_1 *and* S_2 *be two* (H,K)*-composable and deadlock-free systems. Then the composed system* $\mathsf{S}_1 {}^{\mathsf{H} \leftrightarrow \mathsf{K}} \mathsf{S}_2$ *is deadlock-free.*

Given an H-local CFSM M and a participant $\mathsf{K} \in \mathscr{P} \setminus \{\mathsf{H}\}$, call *connecting* a fresh asymmetric sending state of $\mathsf{gw}(M, \mathsf{K})$ whose next outgoing transition does not have K as receiver.

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 ${}_{\mathsf{H}}\lfloor s$ and $s\lfloor_{\mathsf{K}}$ is a deadlock. Without loss of generality, we assume that the deadlock is the left-projection; the case where the deadlock is the right-projection is similar.

First, we show that if a participant A from S_1 has an enabled transition in s then some participant in S_1 has a transition enabled in ${}_{\mathsf{H}}\lfloor s$. Note that ${}_{\mathsf{H}}\lfloor s$ is reachable in S_1 by Proposition B.3.

If $\mathsf{A} \neq \mathsf{H}$ then any transition of A enabled in s is also enabled in ${}_{\mathsf{H}}\lfloor s$ since ${}_{\mathsf{H}}\lfloor s(\mathsf{A}) = s(\mathsf{A})$ by Definition A.1. If $\mathsf{A} = \mathsf{H}$, then either of the following cases occurs

- H has enabled an input

  Assume that the input is from K. Then by construction ${}_{\mathsf{H}}\lfloor s(\mathsf{H})$ has a τ-transition enabled in $\mathsf{S}_1(\mathsf{H})$.

  If the input of H is from a participant A of S_1 then by construction ${}_{\mathsf{H}}\lfloor s(\mathsf{H})$ has an input transition enabled in $\mathsf{S}_1(\mathsf{H})$.

- H has enabled an output.

  Assume that the receiver of such output is K. Then $\mathsf{nof}_{\mathsf{H},\mathsf{K}}(\mathsf{S}_1(\mathsf{H}), s(\mathsf{H})) \asymp \mathsf{nof}_{\mathsf{K},\mathsf{H}}(\mathsf{S}_2(\mathsf{K}), s(\mathsf{K}))$ by Proposition B.3. By definition of nof and of gateway, $\mathsf{nof}_{\mathsf{H},\mathsf{K}}(\mathsf{S}_1(\mathsf{H}), s(\mathsf{H}))$ has an input transition enabled from a participant in S_1, hence $\mathsf{nof}_{\mathsf{K},\mathsf{H}}(\mathsf{S}_2(\mathsf{K}), s(\mathsf{K}))$ has a corresponding output transition enable towards a participant in S_2 by compatibility. By construction (Definition A.1), $s\lfloor_{\mathsf{K}}(\mathsf{K}) = \mathsf{nof}_{\mathsf{K},\mathsf{H}}(\mathsf{S}_2(\mathsf{K}), s(\mathsf{K}))$, hence there is a participant, in particular K, willing to take a transition.

If the receiver of the output from H is a participant of S_1 then, by definition of gateway and configuration projection, we get that in H is willing to perform an output from ${}_{\mathsf{H}}\lfloor s(\mathsf{H})$ in S_1.

- H can perform a τ-transition.

  Then, there is a sequence of transitions of the form $s(\mathsf{H}) \xrightarrow{\tau} \xrightarrow{\mathsf{H}\mathsf{X}!\mathsf{m}}$ in $\mathsf{S}(\mathsf{H})$ with $\mathsf{X} = \mathsf{K}$ or $\mathsf{X} \neq \mathsf{H}$ participant of S_1. If the former case we can reason as in the previous case when H outputs to K. Otherwise, ${}_{\mathsf{H}}\lfloor s(\mathsf{H})$ has an enabled τ-transition in $\mathsf{S}_1(\mathsf{H})$ by Definition A.1.

If s is a deadlock, by definition of deadlock freedom (cf. Definition 2.5), $s \not\to$ but there are participants in S with enabled transitions in s. Under the assumption that $s\lfloor_{\mathsf{K}}$ is deadlock-free, such participants must belong to S_1. By the cases shown above, ${}_{\mathsf{H}}\lfloor s$ enables some participants in S_1; therefore, there is ${}_{\mathsf{H}}\lfloor s \xrightarrow{\lambda}$ because S_1 is deadlock free by hypothesis. It must be that H is involved in all transitions from ${}_{\mathsf{H}}\lfloor s$ of S_1 otherwise $s \xrightarrow{\lambda}$ since for all $\mathsf{X} \in \mathsf{ptp}(\lambda)$ $s(\mathsf{X}) = {}_{\mathsf{H}}\lfloor s(\mathsf{X})$ (by Definition A.1) contrary to our assumption that s is a deadlock. We proceed by case analysis on λ.

$\boxed{\lambda = \mathsf{H}\to\mathsf{X}\colon \mathsf{m}}$ If $\mathsf{X} \neq \mathsf{K}$ then ${}_{\mathsf{H}}\lfloor s(\mathsf{X}) = s(\mathsf{X}) \xrightarrow{\mathsf{H}\mathsf{A}?\mathsf{m}}$; also, H would be in a connecting state and, by Definition A.1, ${}_{\mathsf{H}}\lfloor s(\mathsf{H}) = s(\mathsf{H}) \xrightarrow{\mathsf{H}\mathsf{A}!\mathsf{m}}$. Hence $s \xrightarrow{\lambda}$ contrary to the hypothesis that s is a deadlock configuration.

If $\mathsf{X} = \mathsf{K}$ then we have again a contradiction since $s \xrightarrow{\lambda}$ by Proposition B.3.

$\boxed{\lambda = \mathsf{X}\to\mathsf{H}\colon \mathsf{m}}$ If $\mathsf{X} \neq \mathsf{K}$ then ${}_{\mathsf{H}}\lfloor s(\mathsf{X}) = s(\mathsf{X}) \xrightarrow{\mathsf{A}\mathsf{H}!\mathsf{m}}$ and ${}_{\mathsf{H}}\lfloor s(\mathsf{X}) \xrightarrow{\mathsf{A}\mathsf{H}?\mathsf{m}} = s(\mathsf{H})$ Hence $s \xrightarrow{\lambda}$ contrary to the hypothesis that s is a deadlock configuration.

If $\mathsf{X} = \mathsf{K}$ then we have again a contradiction since $s \xrightarrow{\lambda}$ by Proposition B.3.

$\boxed{\lambda = \tau}$ If $s(\mathsf{H}) = p''$ is fresh then $\mathsf{gw}(\mathsf{S}_1(\mathsf{H}), \mathsf{K})$ must have a sequence of transitions

$$p \xrightarrow{\mathsf{A}\mathsf{H}?\mathsf{m}} p' \xrightarrow{\tau} p'' \xrightarrow{\mathsf{H}\mathsf{K}!\mathsf{m}} r \tag{14}$$

with A participant of S_1 and p' fresh. (Note that it cannot be $s(\mathsf{H}) = p'$ otherwise $s(\mathsf{H}) \xrightarrow{\tau}$ contradicting $s \not\to$.) By Definition A.1, ${}_{\mathsf{H}}\lfloor s(\mathsf{H}) = r$. Hence, by Proposition B.3 we have $s \xrightarrow{\mathsf{H}\to\mathsf{K}\colon \mathsf{m}}$ contradicting $s \not\to$.

If $s(\mathsf{H})$ is not fresh then H has a τ transition enabled at s (because $s(\mathsf{H}) = {}_{\mathsf{H}}\lfloor s(\mathsf{H})$ by Definition A.1) again contradicting $s \not\to$. □

Lemma C.1. *Let* $\mathsf{S} = \mathsf{S}_1 {}^{\mathsf{H}\leftrightarrow\mathsf{K}} \mathsf{S}_2$ *where* S_1 *and* S_2 *are two* (H,K)*-composable systems with* H *and* K *sequential. Given* $s \in \mathscr{R}([\![\mathsf{S}]\!])$*, if*

(1) either ${}_{\mathsf{H}}\lfloor s \xrightarrow{\lambda} s'$ *in* S_1 *and* $\lambda = \tau \implies {}_{\mathsf{H}}\lfloor s(\mathsf{H}) = s'(\mathsf{H})$

(2) or ${}_{\mathsf{H}}\lfloor s \xrightarrow{\tau} \hat{s}$ *involving* H *in* $\mathsf{S}_1(\mathsf{H})$ *and* $\hat{s}$ *reaching a configuration* $\hat{s}'$ *such that* $\hat{s}' \xrightarrow{\lambda} s'$ *in* S_1 *with* $\lambda = \mathsf{H}\to\mathsf{A}\colon \mathsf{m}$

then there is a run $s \xrightarrow{\lambda_1} \cdots \xrightarrow{\lambda_n} \hat{s}$ *in* S *such that* $\lambda_n = \lambda$ *and* ${}_{\mathsf{H}}\lfloor \hat{s} = s'$.

The same holds for the right-projection of S.

Proof. We give the proof for each case.

Case (1) By case analysis on λ noticing that the case $\lambda = \mathsf{H}\to\mathsf{A}\colon \mathsf{m}$ is not possible since ${}_{\mathsf{H}}\lfloor s$ cannot enable output transitions from H by construction (cf. Definition A.1).

$\lambda = \tau$. Then the τ-transition is executed by $\mathsf{A} \neq \mathsf{H}$ in S_1; hence there is a transition ${}_{\mathsf{H}}\lfloor s(\mathsf{A}) \xrightarrow{\tau} q$ in $\mathsf{S}_1(\mathsf{A})$. Observing that $s(\mathsf{A}) = {}_{\mathsf{H}}\lfloor s(\mathsf{A})$ by Definition A.1 we have the thesis since $s \xrightarrow{\tau} s[\mathsf{A} \mapsto q]$.

$\lambda = \mathsf{A}{\to}\mathsf{H}\colon \mathsf{m}$**.** We have

$$ {}_{\mathsf{H}}\lfloor s(\mathsf{H}) \xrightarrow{\mathsf{A}\mathsf{H}?\mathsf{m}} p \text{ in } \mathsf{S}_1(\mathsf{H}) \qquad \text{and} \qquad {}_{\mathsf{H}}\lfloor s(\mathsf{A}) \xrightarrow{\mathsf{A}\mathsf{H}!\mathsf{m}} q \text{ in } \mathsf{S}_1(\mathsf{A}) $$

moreover, $s(\mathsf{H}) = {}_{\mathsf{H}}\lfloor s(\mathsf{H})$ and $s(\mathsf{A}) = {}_{\mathsf{H}}\lfloor s(\mathsf{A})$. Hence,

$$ s \xrightarrow{\mathsf{A}\to\mathsf{H}\colon \mathsf{m}} s'' \qquad \text{with} \qquad s''(\mathsf{X}) = \begin{cases} p, & \text{if } \mathsf{X} = \mathsf{H} \\ q, & \text{if } \mathsf{X} = \mathsf{A} \\ s(\mathsf{X}), & \text{otherwise} \end{cases} $$

by Definition 2.3. Therefore ${}_{\mathsf{H}}\lfloor s'' = s'$.

Case (2) Note that $\mathsf{S}_1(\mathsf{H})$ necessarily contains the transitions ${}_{\mathsf{H}}\lfloor s(\mathsf{H}) \xrightarrow{\tau} \hat{s}'(\mathsf{H}) \xrightarrow{\mathsf{H}\mathsf{A}!\mathsf{m}} r$. Then, by construction, in $\mathsf{gw}(\mathsf{S}_1(\mathsf{H}), \mathsf{K})$ we have

$$ {}_{\mathsf{H}}\lfloor s(\mathsf{H}) \xrightarrow{\mathsf{K}\mathsf{H}?\mathsf{m}} p \xrightarrow{\tau} \hat{s}'(\mathsf{H}) \xrightarrow{\mathsf{H}\mathsf{A}!\mathsf{m}} r \tag{15} $$

note that p and $\hat{s}'(\mathsf{H})$ are fresh, that these are the only transitions from ${}_{\mathsf{H}}\lfloor s(\mathsf{H})$ to r by sequentiality, and that $s(\mathsf{H}) \in \{{}_{\mathsf{H}}\lfloor s(\mathsf{H}), p, \hat{s}'(\mathsf{H})$ by construction (cf. Definition A.1). If $s(\mathsf{H}) = {}_{\mathsf{H}}\lfloor s(\mathsf{H})$ then, by Proposition B.3, we have $s \xrightarrow{\mathsf{K}\to\mathsf{H}\colon \mathsf{m}} \xrightarrow{\tau} s''$ with $s''(\mathsf{H}) = \hat{s}'(\mathsf{H})$. Hence ${}_{\mathsf{H}}\lfloor s'' = {}_{\mathsf{H}}\lfloor s[\mathsf{H} \mapsto r]$ and therefore $s'' \xrightarrow{\mathsf{H}\to\mathsf{A}\colon \mathsf{m}} s''[\mathsf{H} \mapsto r, \mathsf{A} \mapsto s'(\mathsf{A})] = s'''$ which yields the thesis noticing that ${}_{\mathsf{H}}\lfloor s''' = s'$. If $s(\mathsf{H}) = p$ then $s \xrightarrow{\tau} s''$ with $s''(\mathsf{H}) = \hat{s}'(\mathsf{H})$; hence the thesis follows as in the previous case. Finally, if $s(\mathsf{H}) = \hat{s}'(\mathsf{H})$ then $s \xrightarrow{\mathsf{H}\to\mathsf{A}\colon \mathsf{m}} s''$ with $s'' = s[\mathsf{H} \mapsto r, \mathsf{A} \mapsto s'(\mathsf{A})]$ and therefore ${}_{\mathsf{H}}\lfloor s'' = s'$. □

Theorem 5.5 (Lock-freedom preservation)**.** *Let* S_1 *and* S_2 *be two* (H, K)*-composable and lock-free systems with* H *and* K *sequential. Then the composed system* $\mathsf{S}_1 {}^{\mathsf{H}\leftrightarrow\mathsf{K}} \mathsf{S}_2$ *is lock-free.*

Proof. By contradiction, let us assume $\mathsf{S} = \mathsf{S}_1 {}^{\mathsf{H}\leftrightarrow\mathsf{K}} \mathsf{S}_2$ not to be lock-free. Then there is a configuration $s \in \mathscr{R}(\llbracket \mathsf{S} \rrbracket)$ and a participant X not involved in any run from s. Without any loss of generality, we can assume $\mathsf{X} \in \mathsf{S}_1$. We have ${}_{\mathsf{H}}\lfloor s \in \mathscr{R}(\llbracket \mathsf{S}_1 \rrbracket)$ by Proposition B.3 and, by lock-freedom of S_1, ${}_{\mathsf{H}}\lfloor s$ cannot be a lock of S_1 for X. Hence, there exists a run ${}_{\mathsf{H}}\lfloor s \xrightarrow{\lambda_0} s_0 \cdots s_{n-1} \xrightarrow{\lambda_n} s_n$ of S_1 with X involved in λ_n. We show that this induces a run from s in S involving X by induction on n.

- If $n = 0$, by Lemma C.1 there is a run $s \xrightarrow{\psi} \xrightarrow{\lambda} s'$ such that ${}_{\mathsf{H}}\lfloor s' = s_0$ with $\lambda_0 = \lambda$.
- If $n > 0$, we assume that the statement holds for all runs with less than n transitions. If X is involved in λ_i with $0 \le i < n$ then the thesis follows by inductive hypothesis. Let us therefore assume that X is involved in λ_n only. By repeated application of Lemma C.1, there is a run $s \xrightarrow{\psi_1 \cdot \lambda'_1} s'_1 \cdots \xrightarrow{\psi_{n-1} \cdot \lambda'_{n-1}} s'_{n-1} \xrightarrow{\psi_n \cdot \lambda'_n} s'_n$ in S such that $\lambda'_i = \lambda_i$ and ${}_{\mathsf{H}}\lfloor s'_i = s_i$ for each $1 \le i \le n$.

In both cases s reaches a configuration with a run involving X, which contradicts our assumption. □

Theorem 5.6 (Strong lock freedom preservation)**.** *Let* S_1 *and* S_2 *be two* (H, K)*-composable and strongly lock free systems. Then the composed system* $\mathsf{S}_1 {}^{\mathsf{H}\leftrightarrow\mathsf{K}} \mathsf{S}_2$ *is strongly lock free.*

Proof. By contradiction, let us assume $\mathsf{S}_1 {}^{\mathsf{H}\leftrightarrow\mathsf{K}} \mathsf{S}_2$ not to be strongly lock free. This means that there are a reachable configuration s, a participant X, and a maximal run ψ of $\mathsf{S}_1 {}^{\mathsf{H}\leftrightarrow\mathsf{K}} \mathsf{S}_2$ such that $s \xrightarrow{\psi}$ and X is not involved in any of those transitions. By the first part of the proof of Theorem 5.1, there exist two run ${}_{\mathsf{H}}\lfloor s \xrightarrow{\psi_1}$ and $s\lfloor_{\mathsf{K}} \xrightarrow{\psi_2}$ of S_1 and S_2 respectively such that X is involve neither in ψ_1 nor in ψ_2.

- In case ψ is infinite, we get that either ψ_1 or ψ_2 is infinite, and hence maximal.
- In case ψ is finite it is possible to use the second part of the proof of Theorem 5.1 to show that either ψ_1 or ψ_2 is maximal,

In both cases we get a contradiction of the hypothesis that S_1 and S_2 are strong lock free. □