Composition of Synchronous Communicating Systems*

Franco Barbanera[a], Ivan Lanese[b], Emilio Tuosto[c,*]

[a] *Dept. of Mathematics and Computer Science, University of Catania, Catania, Italy*
[b] *Focus Team, University of Bologna/INRIA, Bologna, Italy*
[c] *Gran Sasso Science Institute, L'Aquila, Italy*

Abstract

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. Two communicating systems can be composed by selecting one component per system, and transforming both of them into coupled gateways connecting the two systems. More precisely, a gateway forwards a message received from within its system to the other gateway, which then delivers the message to the other system. Suitable *compatibility* conditions between gateways have been proved sufficient for this composition mechanism to preserve properties such as 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 present paper gives a comprehensive treatment of the case of synchronous communications. We consider both *symmetric synchronous* communications and *asymmetric synchronous* communications (where senders decide independently which message should be exchanged). The composition mechanism preserves different properties under different conditions depending on the considered type of synchronous communication. We show here that preservation of lock freedom requires an additional condition on gateways for asymmetric communication. Such condition is also needed for preservation of deadlock freedom, lock freedom or strong lock freedom for symmetric communications. This is not needed, instead, for preservation of either deadlock freedom or strong lock freedom with asymmetric interactions.

Keywords: Communicating Finite State Machines, Communicating Systems, automata, composition of systems, deadlock freedom, lock freedom, synchronous communications.

1. Introduction

Nowadays applications are increasingly developed for and executed on distributed architectures (e.g., service-oriented architectures, microservices, cloud, etc.). Communication is therefore an essential constitutive element of modern software. As a matter of fact, APIs, libraries, and languages feature different

* 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 "National Center for "HPC, Big Data e Quantum Computing", Programma M4C2 – dalla ricerca all'impresa – Investimento 1.3: Creazione di "Partenariati estesi alle università, ai centri di ricerca, alle aziende per il finanziamento di progetti di ricerca di base" –Next Generation EU. The authors have also been partially supported by INdAM as members of GNCS (Gruppo Nazionale per il Calcolo Scientifico). The authors acknowledge the support of the PRO3 MUR project Software Quality, and PNRR MUR project VITALITY (ECS00000041), Spoke 2 ASTRA - Advanced Space Technologies and Research Alliance. The authors thank the reviewers of the conference and present versions of this paper for their helpful comments. Finally the authors thank Mariangiola Dezani for her support.

*Corresponding author

Email addresses: barba@dmi.unict.it (Franco Barbanera), ivan.lanese@gmail.com (Ivan Lanese), emilio.tuosto@gssi.it (Emilio Tuosto)

communication mechanisms. Several models, such as process algebras, transition systems, Petri nets, automata, etc., have been proposed to study interactions between systems. Indeed, developing and reasoning on communicating systems are difficult endeavours requiring to juggle the *what* has to be communicated (a.k.a. *business logic*) with the *how* information spreads across a system, sometimes called *application level protocol*. In fact, conceptual and programming errors may occur in the realisation of application level protocols. Typical errors occur when a system is not guaranteed to enjoy some relevant communication properties, like those we consider in the present paper, namely *deadlock-freedom*, *lock-freedom* and *strong lock-freedom*. Roughly, deadlock-freedom guarantees a system to get stuck only if all of its components terminated; lock-freedom guarantees any component willing to communicate to eventually do so in one of the possible continuations of the system; in a strong lock-free system, instead, each component willing to communicate will eventually do so in any possible continuation of the system. Such communication properties may fail when a system makes a choice among different alternatives, and components have inconsistent "views" of which alternative has been selected. If this happens, some components may reach a state no longer "compatible" with the state of their partners and therefore communications do not happen as expected. Let us discuss deadlock freedom through a simple example (similar examples can be given for lock freedom or strong 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 appeal to CCS [1] to give possible specifications of this scenario. Let us start with the following CCS[1] agents:

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

where C and S respectively give the behaviour of clients and servers. It is a simple observation that after the handshake on r the system $\mathsf{C} \mid \mathsf{S}$, where C and S 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.

Enforcing correctness of communicating systems is notoriously difficult. A reason for this difficulty is that it is hard to define composition mechanisms that preserve the communication properties mentioned above.

Contributions. Driven by the motivations in Section 2, we investigate the mechanism of composition by gateways introduced in [2, 3] (for asynchronous communications) in the setting of communicating finite-state machines (CFSMs) [4] with synchronous communications.

Intuitively, composition by gateways transforms two participants H and K, taken from two separate systems, into coupled gateways used to connect the two systems. More precisely, when H receives a message from its own system, it forwards it to K to be delivered to the other system and vice versa. Our investigation concerns deadlock-, lock-, and strong lock-freedom preservation under a notion of composability. This notion combines compatibility, which intuitively prescribes H and K to exhibit dual behaviour, namely if H sends some message then K is willing to receive it and vice versa, with further technical restrictions.

This paper integrates and refines the preliminary results in [5] and some of the ones in [6]. The latter also considers an additional form of composition, called semi-direct composition, which we do not take into account here. We consider both the symmetric and asymmetric communication models. Roughly, in a (one-to-one) symmetric communication model, sender and receiver in an interaction equally concur to determining the message to be exchanged. In the asymmetric model, instead, such a symmetry is broken by a bias towards the sender, which autonomously decides the message to be sent and the intended receiver. Table 1 summarises the results of this paper and refers to their preliminary conference versions (if any). We show that

[1] In CCS, *agents* (or *processes*) interact by synchronising on *communication ports*; an output action on a given port x, is denoted with $\bar{\mathsf{x}}$ while an input action on x is denoted by x itself. Formally, CCS can be defined as an algebra (of agents) whose operators act on ports and agents itself. A fundamental operator is the *non-deterministic choice* $\_ + \_$ which acts on agents: $\mathsf{P} + \mathsf{Q}$ represents an agent that can non-deterministically behave as P or as Q. Other important operators are prefixes $\alpha.\_$ where α is a communication action: the agent $\alpha.\mathsf{P}$ behaves as P after the execution of the communication α.

		Deadlock-freedom	Lock-freedom	Strong lock-freedom
symm. case	Composability	✗ [6] and Ex. 6.7	✗ Ex. 6.7	✗ Ex. 6.7
	Comp. + Seq.	✓ [6] and Thm. 8.7	✓ Thm. 8.4	✓ Thm. 8.8
asymm. case	Composability	✓ [5] and Thm. 6.2	✗ Ex. 6.6	✓ [5] and Thm. 6.4
	Comp. + Seq.	✓ trivial from above	✓ [5] and Thm. 8.4	✓ trivial from above

Table 1: Properties preserved by composition

in the asymmetric case composition by gateways (i) preserves deadlock freedom (Theorem 6.2), as well as a strong version of lock freedom (Theorem 6.4), provided that systems are *composable*; and (ii) preserves lock freedom (Theorem 8.4) if systems are composable and gateways are *sequential*[2], namely each state has at most one outgoing transition;

in the symmetric case composition by gateways preserves all the communication properties we take into account –i.e., deadlock-, lock- and strong lock-freedom (Theorem 8.7, Theorem 8.4 and Theorem 8.8, respectively)– whenever the compatibility condition is strenghtened with the sequentiality requirement.

Interestingly, preservation of deadlock freedom and strong lock-freedom in the asymmetric case can be guaranteed under milder conditions than in the symmetric case. We give counterexamples showing that properties are not preserved under even weaker conditions.

As reported in Table 1, the results concerning lock- and strong lock-freedom in the symmetric case are new. For the others, besides an improved presentation, we provide full and completely revised proofs. The definition of compatibility (Definition 4.6) has been modified with respect to [5] in order to fix some formalisation issues.

Structure of the paper. Section 2 discusses our motivations. Section 3 introduces systems of (asymmetric synchronous) CFSMs, their basics, and their communication properties. Composition by gateways is introduced and discussed in Section 4, together with the compatibility relation which is at the basis of properties preservation. Section 5 relates computations in the component systems to computations in the resulting composition, a relation needed for our main results. Section 6 is devoted to the preservation of communication properties if only composability is considered. In Section 7 we show that composability is needed for the results above. By extending composability with the further requirement of sequentiality, in Section 8 we get all the other results of properties preservation. Conclusions, related and future work are discussed in Section 9. For the sake of readability, some auxiliary results needed for proving the preservation results are in Appendix A, in Appendix B, and in Appendix C.

2. Motivations

We motivate our interest in asymmetric synchronous communications by considering again the deadlock issue of the client-server example in Section 1. The problem with the agents in (1) is that the choice of what communication should happen after a request is non-deterministically taken independently by C and S instead of letting S to take the decision and drive C "on the right" branch. Let us consider the following alternative agents:

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

In an asynchronous setting the server S in (2) decides what to reply to the client C after the request has been made; C becomes aware of the choice through the signal received on one of the two ports. The situation is

[2]The term 'sequential' refers here to the absence of both concurrency and choices.

radically different when communication is synchronous *and* symmetric. In fact, in such case the resolution of the choice hinges on the communication mechanism: a branch is taken as soon as C and S synchronise on port a or on port u. The resolution of the choice is therefore *external*, in the jargon of [7]. Basically this means that agents cannot decide which branch to take *in isolation* (as in the asynchronous case): the symmetry in the communication implies that parties equally contribute to the decision on how to continue the computation. As noted in [7], an alternative way to encompass non-determinism in CCS-like languages is to get rid of internal actions from the syntax and "split" the non-determistic choice operator into two: an operator for *internal* choices and one for *external* choices (respectively denoted as $\_ \oplus \_$ and $\_ \Box \_$ in [7]). This breaks the symmetry between input and output of synchronous communication; now the choice is entirely resolved on "one side" because the sender must select the branch to fire "before" executing the output action. In [7] this is formally specified by making agents of the form $\bar{\mathsf{a}}.\mathsf{P} + \mathsf{P}'$ to "internally" reduce to $\bar{\mathsf{a}}.\mathsf{P}$ so that the parallel composition of a sender and a receiver, say $\bar{\mathsf{a}}.\mathsf{P} \mid (\mathsf{a}.\mathsf{Q} + \mathsf{Q}')$ can reduce to $\mathsf{P}|\mathsf{Q}$. This semantics provides an abstract model of *asymmetric* synchronous communications, where choices on outputs are "internally" resolved making the sender decide what to communicate and to which partner. Interestingly, internal and external choice operators are at the core of behavioural type systems [8] and behavioural contracts (used, e.g., in [9, 10, 11]).

Besides the theoretical motivation above, asymmetric synchronous communications have also a more practical motivation. In fact, asymmetric synchronous communications abstract a rather common programming pattern where a component chooses the message to send as well as the target of the communication *depending* on some condition on its own state, e.g., using a conditional.

We now discuss the interplay between communications and compositionality by elaborating on our client-server example. Assume that S acts as a proxy to another server, say S'. If the server S cannot return an answer to C and C is a privileged client, S interacts with S' on port p. Answers are sent directly to C if S' can compute them, otherwise S' returns an unknown on port u' to S which forwards it to C. This is modelled by the agents

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

Note that this change is completely transparent to agent C, which in fact stays as in (2). Here S resolves the choice and then engages in the corresponding communication.

It is interesting to note that it is now more difficult to ascertain if these choices may lead to a deadlock than for the agents in (2). In fact, the decision of S may involve also S'. The parallel composition of the client C from (2) with the agents in (3) may indeed deadlock because, when C and S synchronise on port a, S' hangs on port p and, likewise, if C and S' synchronise on port a then S hangs on port u'.

The above approach has been applied in [2, 3] for the well-known model of systems of *communicating finite-state machines* (CFSMs) [4], that is sets of finite-state automata capable of exchanging messages. 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 dual behaviours: gateway H is able to receive a message whenever gateway K is willing to send one and vice versa.

The results in [2, 3] are developed for the asynchronous semantics of CFSMs. These results have been transferred in [6] to a setting where CFSMs communicate synchronously much like 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, which, in absence of internal actions, amount to have only *external* or *global* non-determinism [12].

3. Communicating Finite State Machines with Internal Actions

In this section we define a natural generalisation of Communicating Finite State Machines [4] suitable for formalising the notions of both symmetric and asymmetric synchronous interactions, as well as the composition of systems via gateways.

We begin by recalling a few basic notions on Finite State Automata (FSAs).
A *finite state automaton* (FSA) is a tuple $A = \langle \mathcal{S}, q_0, \mathcal{L}, \rightarrow \rangle$ where

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

Hereafter, we let λ range over $\mathcal{L} \cup \{\tau\}$. As usual $q \xrightarrow{\lambda} q'$ abbreviates $(q, \lambda, q') \in \rightarrow$. We say that $q \xrightarrow{\lambda} q'$ is an *outgoing* (resp. *incoming*) transition of q (resp. q'). Also, $q \rightarrow$ states that q has an outgoing transition, and $q \not\rightarrow$ states that there are no transitions from q. Let $\_\cdot\_$ be the concatenation operation on sequences of labels and write $p \xrightarrow{\pi} q$ where $\pi = \lambda_1 \cdot \lambda_2 \cdot \ldots \cdot \lambda_n$ (here we identify labels with sequences of labels of length one) whenever $p \xrightarrow{\lambda_1} p_1 \xrightarrow{\lambda_2} \ldots \xrightarrow{\lambda_n} p_n = q$. We let $\pi, \psi, \ldots$ range over $(\mathcal{L} \cup \{\tau\})^\star$ (i.e., sequences of labels) and define the set of *states in A reachable from q* as

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

The set of *reachable states in A* is $\mathcal{R}(A) = \mathcal{R}(A, q_0)$. 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. As in [4], we formalise communicating systems as FSAs where accepting states are disregarded. The definition of CFSMs in [4] is extended here with τ labels to account for asymmetric synchronisation. 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. An *action label* is either an output or an input label. An *output label* is written as $\mathsf{A}\,\mathsf{B}!\mathsf{m}$ and represents the willingness of A to send message m to B; likewise, an *input label* is written as $\mathsf{A}\,\mathsf{B}?\mathsf{m}$ and represents the willingness of B to receive message m from A. Interaction labels, instead, have the form $\mathsf{A}{\rightarrow}\mathsf{B}\colon \mathsf{m}$, which represents the interaction where the message m is sent from A to B, and received from the latter. The *subject* of an output label $\mathsf{A}\,\mathsf{B}!\mathsf{m}$ and of an input label $\mathsf{B}\,\mathsf{A}?\mathsf{m}$ is A. We also say that A (resp. B) is the *sender* (resp. *receiver*) in the interaction label $\mathsf{A}{\rightarrow}\mathsf{B}\colon \mathsf{m}$.

The set of action- and interaction-labels are hence defined by

$$\begin{aligned}\mathcal{L}_{\text{act}} &= \{\mathsf{A}\,\mathsf{B}!\mathsf{m}, \mathsf{A}\,\mathsf{B}?\mathsf{m} \mid \mathsf{A} \neq \mathsf{B} \in \mathfrak{P} \quad \text{and} \quad \mathsf{m} \in \mathcal{M}\} \\ \mathcal{L}_{\text{int}} &= \{\mathsf{A}{\rightarrow}\mathsf{B}\colon \mathsf{m} \mid \mathsf{A} \neq \mathsf{B} \in \mathfrak{P} \quad \text{and} \quad \mathsf{m} \in \mathcal{M}\}\end{aligned}$$

In the following, we use function $\mathsf{ptp}(\lambda)$ computing the set of participants occurring in a label λ, and defined as $\mathsf{ptp}(\tau) = \emptyset$ and $\mathsf{ptp}(\mathsf{A}{\rightarrow}\mathsf{B}\colon \mathsf{m}) = \mathsf{ptp}(\mathsf{A}\,\mathsf{B}!\mathsf{m}) = \mathsf{ptp}(\mathsf{A}\,\mathsf{B}?\mathsf{m}) = \{\,\mathsf{A}, \mathsf{B}\,\}$ and, for a sequence $\pi = \lambda_1 \cdot \ldots \cdot \lambda_n$, we let $\mathsf{ptp}(\pi) = \cup_{1 \leq i \leq n} \mathsf{ptp}(\lambda_i)$.

Definition 3.1 (τ-CFSM)**.** *A τ-CFSM is an FSA with labels in $\mathcal{L}_{act} \cup \{\,\tau\,\}$. A τ-CFSM is* A-local *if all its non τ-transitions have subject A.*

An output *(resp.* input*) transition is a transition labelled by an output (resp. input) label.*

We now define systems of τ-CFSM.

Definition 3.2 (τ-systems)**.** *Let $\mathcal{P} \subseteq \mathfrak{P}$ be finite set of participants. A τ-system over $\mathcal{P}$ is a map $\mathsf{S} = (M_{\mathsf{A}})_{\mathsf{A} \in \mathcal{P}}$ assigning an A-local τ-CFSM M_{A} to each participant $\mathsf{A} \in \mathcal{P}$ where any participant occurring in a transition of M_{A} is in $\mathcal{P}$.*

A τ-system consists of *named* τ-CFSMs, which we formally describe as a function from $\mathcal{P}$ (the names of the participants of the system) to the set of τ-CFSMs. A τ-CFSM with name A (i.e., describing the behaviour of participant A) is naturally requested to be A-local, that is A is the sender of its output actions and the receiver of its input ones. Note that Definition 3.2 requires also that any input or output label

does refer to participants belonging to the system itself. In other words, Definition 3.2 restricts to *closed* τ-systems.

We define the synchronous semantics of τ-systems as an FSA (differently from the asynchronous case, where the set of states can be infinite). Hereafter, $\mathrm{dom}(f)$ denotes the domain of a function f and $f[x \mapsto y]$ denotes the update of f in a point $x \in \mathrm{dom}(f)$ with the value y.

Definition 3.3 (Synchronous semantics)**.** *Let* S *be a* τ*-system. The* synchronous semantics *of* S *is the FSA* $[\![\mathsf{S}]\!] = \langle \mathcal{S}, s_0, \mathcal{L}_{int} \cup \{\tau\}, \rightarrow \rangle$ *where*

- $\mathcal{S}$ *is the set of synchronous* configurations *of* S*, where a configuration 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* $\mathsf{A} \in \mathrm{dom}(\mathsf{S})$*;*
- $s_0 = (q_{0\mathsf{A}})_{\mathsf{A} \in \mathrm{dom}(\mathsf{S})} \in \mathcal{S}$ *is the* initial *configuration where, for each* $\mathsf{A} \in \mathrm{dom}(\mathsf{S})$*,* $q_{0\mathsf{A}}$ *is the initial state of* $\mathsf{S}(\mathsf{A})$*;*
- $s \xrightarrow{\tau} s[\mathsf{A} \mapsto q] \in [\![\mathsf{S}]\!]$ *if* $s(\mathsf{A}) \xrightarrow{\tau} q \in \mathsf{S}(\mathsf{A})$*;*
- $s \xrightarrow{\mathsf{A} \to \mathsf{B} : \mathsf{m}} s[\mathsf{A} \mapsto q, \mathsf{B} \mapsto q'] \in [\![\mathsf{S}]\!]$ *if* $s(\mathsf{A}) \xrightarrow{\mathsf{A}\,\mathsf{B}!\mathsf{m}} q \in \mathsf{S}(\mathsf{A})$ *and* $s(\mathsf{B}) \xrightarrow{\mathsf{A}\,\mathsf{B}?\mathsf{m}} q' \in \mathsf{S}(\mathsf{B})$*.*

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

A participant $\mathsf{A} \in \mathcal{P}$ *is* involved in a transition $s \xrightarrow{\lambda} s'$ *if either* $\mathsf{A} \in \mathit{ptp}(\lambda)$ *or* $\lambda = \tau$*,* $s(\mathsf{A}) \xrightarrow{\tau} q$ *in* $\mathsf{S}(\mathsf{A})$*, and* $s' = s[\mathsf{A} \mapsto q]$*;* A *is* involved in a run *if* A *is involved in one of its transitions.*

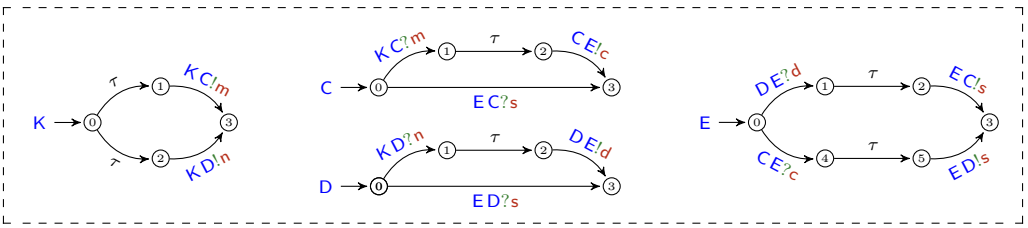
As expected, an interaction $\mathsf{A} \to \mathsf{B} : \mathsf{m}$ occurs when A performs an output $\mathsf{A}\,\mathsf{B}!\mathsf{m}$ and B the corresponding input $\mathsf{A}\,\mathsf{B}?\mathsf{m}$.

A τ-CFSM without τ-transitions is a CFSM in the sense of [4]; following [4], we dub *communicating systems* those τ-systems consisting of CFSMs only. We simply use *system* when the type of machines is immaterial. Also, the symmetric synchronous semantics in [6] for systems without τ-transitions can be readily obtained from the above definition by disregarding the clause for τ-transitions. It is worth remarking that the use of τ-transitions in the above definition is not equivalent to that of buffers of size 1 in the model of asynchronous CFSMs [4], since we are modelling (asymmetric) synchronous interactions, which are *blocking*.

We cast in our framework the usual definition of runs.

Definition 3.4 (Runs)**.** *A* run from s *is a (possibly infinite) sequence* $\xi = s \xrightarrow{\lambda_1} s_1 \cdots s_{n-1} \xrightarrow{\lambda_n} s_n \cdots$ *starting from* s*. A run is* maximal *if it is infinite or it reaches a configuration* s' *such that* $s' \not\rightarrow$*. A run* ξ *is* fair *if whenever there is an interaction label* λ *and an index* $j \geq 1$ *such that* $s_i \xrightarrow{\tau^\star} \xrightarrow{\lambda} \in [\![\mathsf{S}]\!]$ *for infinitely-many indexes* $i \geq j$*, we have that for each* X *involved in* λ *there is* $h \geq j$ *such that* $s_h \xrightarrow{\lambda'} s_{h+1}$ *with* X *involved in* λ'*. A* run of S *is a run from the initial configuration of* S*.*

Example 3.5 (Runs)**.** *Let us consider the* τ*-system* $\mathsf{S} = (M_{\mathsf{X}})_{\mathsf{X} \in \{\mathsf{K},\mathsf{C},\mathsf{D},\mathsf{E}\}}$*, where*



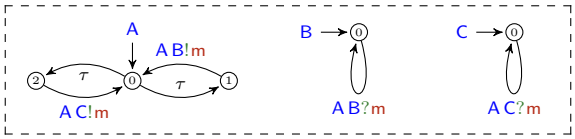
are the τ*-CFSM*[3] M_{K}*,* M_{C}*,* M_{D}*, and* M_{E} *respectively. Then*

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

[3] Hereafter, we specify communicating systems with the graphical notation introduced in Example 3.5.

is a sequence of transitions of $[\![\mathsf{S}]\!]$ *starting from* s_0 *induced by Definition 3.3. Note that this is a maximal run of* S. ◇

Example 3.6 (Fair runs)**.** *Let us consider the* τ*-system* $\mathsf{S} = (M_{\mathsf{X}})_{\mathsf{X}\in\{\mathsf{A},\mathsf{B},\mathsf{C}\}}$*, where*



Then the following infinite run of S *is fair*

$$\begin{array}{rcccc} s_0 = (0_{\mathsf{A}}, 0_{\mathsf{B}}, 0_{\mathsf{C}}) & \xrightarrow{\tau} & (1_{\mathsf{A}}, 0_{\mathsf{B}}, 0_{\mathsf{C}}) & \xrightarrow{\mathsf{A}\to\mathsf{B}:\,\mathsf{m}} & (0_{\mathsf{A}}, 0_{\mathsf{B}}, 0_{\mathsf{C}}) \\ & \xrightarrow{\tau} & (2_{\mathsf{A}}, 0_{\mathsf{B}}, 0_{\mathsf{C}}) & \xrightarrow{\mathsf{A}\to\mathsf{C}:\,\mathsf{m}} & (0_{\mathsf{A}}, 0_{\mathsf{B}}, 0_{\mathsf{C}}) \\ & \xrightarrow{\tau} & (1_{\mathsf{A}}, 0_{\mathsf{B}}, 0_{\mathsf{C}}) & \xrightarrow{\mathsf{A}\to\mathsf{B}:\,\mathsf{m}} & (0_{\mathsf{A}}, 0_{\mathsf{B}}, 0_{\mathsf{C}}) \\ & \xrightarrow{\tau} & (2_{\mathsf{A}}, 0_{\mathsf{B}}, 0_{\mathsf{C}}) & \xrightarrow{\mathsf{A}\to\mathsf{C}:\,\mathsf{m}} & (0_{\mathsf{A}}, 0_{\mathsf{B}}, 0_{\mathsf{C}}) \\ & \xrightarrow{\tau} & \dots & & \end{array}$$

The following infinite run of S*, instead, is not fair.*

$$\begin{array}{rcccc} s_0 = (0_{\mathsf{A}}, 0_{\mathsf{B}}, 0_{\mathsf{C}}) & \xrightarrow{\tau} & (1_{\mathsf{A}}, 0_{\mathsf{B}}, 0_{\mathsf{C}}) & \xrightarrow{\mathsf{A}\to\mathsf{B}:\,\mathsf{m}} & (0_{\mathsf{A}}, 0_{\mathsf{B}}, 0_{\mathsf{C}}) \\ & \xrightarrow{\tau} & (1_{\mathsf{A}}, 0_{\mathsf{B}}, 0_{\mathsf{C}}) & \xrightarrow{\mathsf{A}\to\mathsf{B}:\,\mathsf{m}} & (0_{\mathsf{A}}, 0_{\mathsf{B}}, 0_{\mathsf{C}}) \\ & \xrightarrow{\tau} & \dots & & \end{array}$$

The run above is not fair for C *since there are infinitely many configurations where the transition* $\mathsf{A}\to\mathsf{C}:\mathsf{m}$ *could be executed (after the* τ*-transition to state* 2 *in* A*), however,* A *keeps choosing the* τ*-transition to state* 1*, so preventing* C *to execute.* ◇

3.1. Asymmetric Synchronous Interactions

We model asymmetric synchronous communications by restricting the (too liberal) τ-CFSMs formalism.

Definition 3.7 (τ!-CFSM)**.** *A* τ!-CFSM *(pronounced 'tau-bang CFSM') is a* τ*-CFSM, say* M*, such that*

- *for all output transitions* $p \xrightarrow{\lambda} q$ *of* M*,* p *has exactly one incoming transition in* M*, and such transition is labelled by* τ*;*
- *for all* τ*-transitions* $p \xrightarrow{\tau} q$ *of* M*,* q *has exactly one outgoing transition in* M*, and such transition is an output transition.*

We call τ!*-systems* τ*-systems consisting of* τ!*-CFSMs only.*

Observe that both conditions in the above definition imply $p \neq q$. All our proofs could easily be adapted to an alternative definition of τ!-CFSM, where the condition of having τ-transitions prefixing output transitions like $p \xrightarrow{\mathsf{X}\,\mathsf{Y}!\mathsf{z}} q$ is dropped if $p \xrightarrow{\mathsf{X}\,\mathsf{Y}!\mathsf{z}} q$ is the only outgoing transition from p. Indeed, in this case no actual choice about which output to perform needs to be taken. We however prefer to stick to Definition 3.7 because (i) our uniform treatment of transitions allows us to immediately adapt definitions to more abstract settings and (ii) uniformity allows us to simplify some technicalities.

Example 3.8. *The* τ*-CFSMs of Example 3.5 are* τ!*-CFSMs.*

3.2. Communication Properties

As discussed in Section 1, we shall study the preservation of communication properties under composition. We shall consider the following properties: deadlock freedom, lock freedom, and strong lock freedom. The definitions below adapt the ones in [13] to a synchronous setting (as done also in [14, 15, 6]).

Definition 3.9 (Communication properties)**.** *Let* S *be a system on* $\mathcal{P}$*.*

Deadlock freedom. *A configuration* $s \in \mathcal{R}(\llbracket \mathsf{S} \rrbracket)$ *is a* deadlock *if*

- *there exists* $\mathsf{A} \in \mathcal{P}$ *such that* s *enables* A *in* S *and*
- s *has no outgoing transitions in* $\llbracket \mathsf{S} \rrbracket$*.*

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

Lock freedom. *A configuration* $s \in \mathcal{R}(\llbracket \mathsf{S} \rrbracket)$ *is a* lock *for* $\mathsf{A} \in \mathcal{P}$ *if*

- s *enables* A *in* S *and*
- *there is no run from* s *involving* A*.*

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

Strong lock freedom. *A configuration* $s \in \mathcal{R}(\llbracket \mathsf{S} \rrbracket)$ *is a* weak lock *for* $\mathsf{A} \in \mathcal{P}$ *if*

- s *enables* A *in* S *and*
- *there is at least a maximal fair run from* s *not involving* A*.*

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

Proposition 3.10. *Strong lock freedom implies lock freedom and lock freedom implies deadlock freedom.*

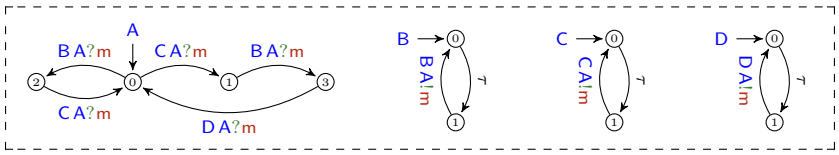
Proof. Directly from the definitions. □

Example 3.11. *Let us consider the system* S *of Example 3.5. The only other maximal run in* $\llbracket \mathsf{S} \rrbracket$ *starting from* s_0*, besides the one described in Example 3.5, is*

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

Then $\mathcal{R}(\llbracket \mathsf{S} \rrbracket)$ *consists of all and only the configurations on these two runs. By inspecting these finitely many configurations we can check that* S *is strongly lock free.*

Let us consider now the system $\mathsf{S} = (M_\mathsf{X})_{\mathsf{X} \in \{\mathsf{A},\mathsf{B},\mathsf{C},\mathsf{D}\}}$*, where*



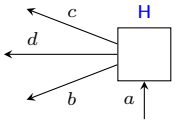
Such a system is both deadlock- and lock-free, but it is not strongly lock-free since its initial configuration s_0 *is a weak-lock for* D*. In fact it enables* D*, but the following infinite run*

$$\begin{array}{rcl@{\quad}c@{\quad}l} s_0 = (0_\mathsf{A}, 0_\mathsf{B}, 0_\mathsf{C}, 0_\mathsf{D}) & \xrightarrow{\tau} & (0_\mathsf{A}, 1_\mathsf{B}, 0_\mathsf{C}, 0_\mathsf{D}) & \xrightarrow{\mathsf{B}\to\mathsf{A}:\,\mathsf{m}} & (2_\mathsf{A}, 0_\mathsf{B}, 0_\mathsf{C}, 0_\mathsf{D}) \\ & \xrightarrow{\tau} & (2_\mathsf{A}, 0_\mathsf{B}, 1_\mathsf{C}, 0_\mathsf{D}) & \xrightarrow{\mathsf{C}\to\mathsf{A}:\,\mathsf{m}} & (0_\mathsf{A}, 0_\mathsf{B}, 0_\mathsf{C}, 0_\mathsf{D}) \\ & \xrightarrow{\tau} & \cdots & & \end{array}$$

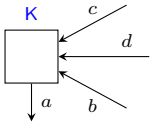
does not involve D *but it is fair because there is no interaction involving* D *which is enabled infinitely many times (possibly after some* τ*-transitions). Actually, no interaction involving* D *is ever enabled.* ◇

4. Composition via Gateways

Recently, an approach to the composition of concurrent and distributed systems has been proposed in [2]. The composition mechanism is based on the idea that two given systems, say S_1 and S_2, are composed by transforming two participants, one per system, say H in S_1 and K in S_2, into "coupled forwarders".

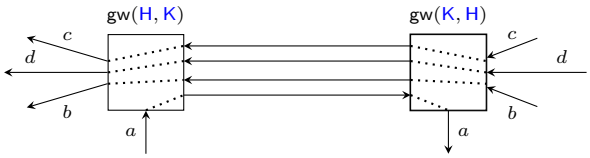


Basically, each message that H receives from a participant in S_1 is forwarded to K and vice versa. In order to clarify the underlying idea, let us consider a system S_1 made of several communicating participants, among which a participant H (graphically represented on the left) is present. Participant H can receive messages of type a from other participants of S_1 and send messages of type b, c, or d. For the purpose of this example, we abstract from the way communications are performed and from the logical order of the exchanged messages. Now let us consider a second system S_2, where a participant K is present, and assume that K (graphically represented on the right) can send to other participants messages of type a, and receive messages of type b, c, or d.



Participants H and K are *compatible* in the sense that, if some participant of the system S_1 needs a message "produced" by H, this message can actually be the one received in system S_2 by K. Symmetrically, the messages that H receives from other participants of S_1 could be forwarded to S_2 as if they were messages sent by K to participants of S_2. From this point of view, the behaviours of H and K can be looked at as "interfaces" which describe what one system expects from the other in case they interact. As observed in [2, 16, 3], a remarkable feature of such an approach is that it enables the composition of systems originally designed as *closed*. Provided that two compatible participants can be found, any two systems can be composed by transforming as sketched above the two selected participants.

The composition mechanism consists then in replacing the two participants H and K by two forwarders, $\mathsf{gw}(\mathsf{H},\mathsf{K})$ and $\mathsf{gw}(\mathsf{K},\mathsf{H})$, respectively, and connecting them as follows:



Hereafter, H and K denote the participant selected for the composition in, respectively, S_1 and S_2.

Using gateways, open systems can hence be seen as a special kind of closed ones, satisfying suitable compatibility conditions when composing them into a whole; so exploiting the flexibility of modular development of systems also in case these were not developed as open.

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.

Following the intuition above, we now define composition via gateways for both systems of CFSMs and of τ!-CFSMs. They slightly differ since we have to make sure that a gateway obtained out of a CFSM (resp. a τ!-CFSM) is still a CFSM (resp. a τ!-CFSM).

Definition 4.1 (Gateways)**.** *Fix two distinct participants* $\mathsf{H},\mathsf{K} \in \mathcal{P}$*.*

i) *The* gateway towards K of an H-local CFSM M *is the CFSM* $\mathsf{gw}(M,\mathsf{K})$ *obtained by replacing in* M*:*

- *each transition* $q \xrightarrow{\mathsf{H\,A!m}} r \in M$ *with*

$$q \xrightarrow{\mathsf{K\,H?m}} q' \xrightarrow{\mathsf{H\,A!m}} r \qquad \text{for some fresh state } q' \tag{4}$$

- *each transition* $q \xrightarrow{\mathsf{A\,H?m}} r \in M$ *with*

$$q \xrightarrow{\mathsf{A\,H?m}} q' \xrightarrow{\mathsf{H\,K!m}} r \qquad \text{for some fresh state } q' \tag{5}$$

ii) *The* gateway towards K of an H-local τ!-CFSM M *is the* τ!*-CFSM* $\mathsf{gw}(M, \mathsf{K})$ *obtained by replacing in* M*:*

- *each pair of consecutive transitions* $q \xrightarrow{\tau} q' \xrightarrow{\mathsf{H}\,\mathsf{A}!\mathsf{m}} r$ *with*

$$q \xrightarrow{\mathsf{K}\,\mathsf{H}?\mathsf{m}} q'' \xrightarrow{\tau} q' \xrightarrow{\mathsf{H}\,\mathsf{A}!\mathsf{m}} r \qquad \text{for some fresh state } q'' \tag{6}$$

- *each transition* $q \xrightarrow{\mathsf{A}\,\mathsf{H}?\mathsf{m}} r$ *with*

$$q \xrightarrow{\mathsf{A}\,\mathsf{H}?\mathsf{m}} q' \xrightarrow{\tau} q'' \xrightarrow{\mathsf{H}\,\mathsf{K}!\mathsf{m}} r \qquad \text{for some fresh states } q' \text{ and } q'' \tag{7}$$

We call the sequences of transitions (4) *and* (6) q-r importing segments (from K) *and the sequences of transitions* (5) *and* (7) q-r exporting segments (to K). *Also, we call* internal *the state* q' *in* (4) *and* (5) *as well as the states* q' *and* q'' *in* (6) *and* (7).

Note that a gateway $\mathsf{gw}(M, \_)$ executes segments of the form described above which are different for CFSMs and τ!-CFSMs; indeed our notation is overloaded, but contexts will clarify which type of machines is involved. Also, by very construction, we have the following

Fact 4.2. *Given an* H*-local* τ!*-CFSM* M *and a participant* K *not occurring in* M*, each state of* $\mathsf{gw}(M, \mathsf{K})$ *has at most one incoming or outgoing* τ *transition.*

As a consequence of the above fact, the alternative definition of τ!-CFSMs discussed after Definition 3.7 would make gateways for τ!-CFSM identical to gateways for CFSMs.

We compose systems with disjoint participants 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 $\mathrm{dom}(g)$.

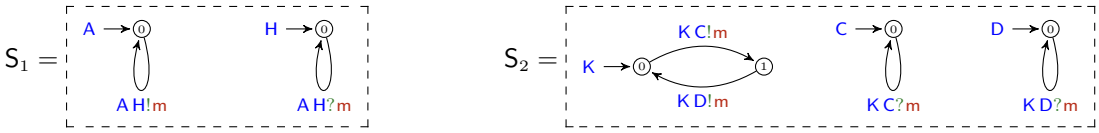
Definition 4.3 (System composition)**.** *Let* S_1 *and* S_2 *be either communicating systems or* τ!*-systems with disjoint domains. The* composition of S_1 and S_2 via $\mathsf{H} \in \mathrm{dom}(\mathsf{S}_1)$ and $\mathsf{K} \in \mathrm{dom}(\mathsf{S}_2)$ *is defined as*

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

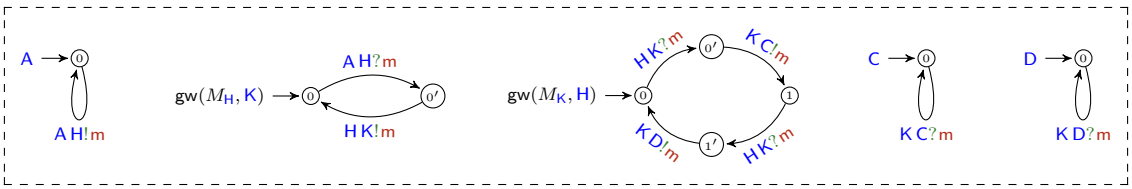
(Note that $\mathrm{dom}(\mathsf{S}_1 {}^{\mathsf{H} \leftrightarrow \mathsf{K}} \mathsf{S}_2) = \mathrm{dom}(\mathsf{S}_1) \cup \mathrm{dom}(\mathsf{S}_2)$*.)*

We remark 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 two suitable participants in the "to-be-connected" systems and transforming them into two forwarders. Also, note that the notion of composition above is structural: corresponding notions of behavioural composition have been studied in [16] for multiparty session types.

Example 4.4. *Take the communicating systems* S_1 *and* S_2 *below*

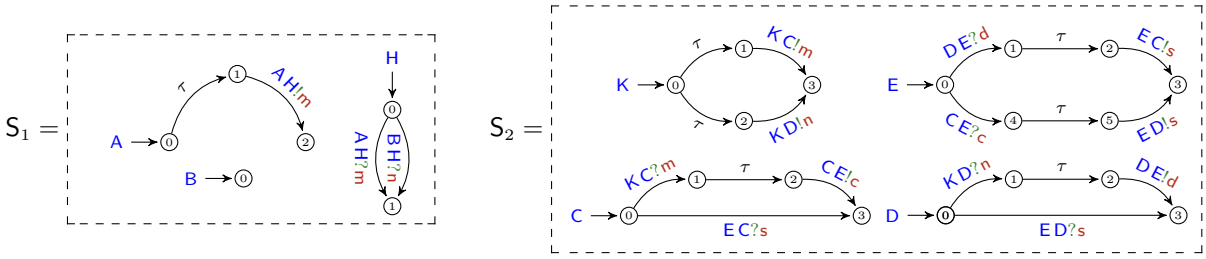


The following communicating system

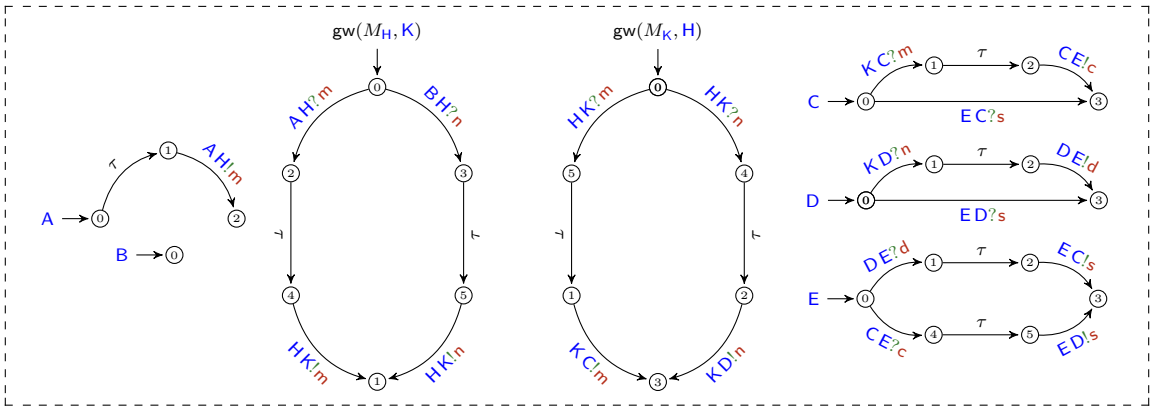


is the composition $\mathsf{S}_1 {}^{\mathsf{H} \leftrightarrow \mathsf{K}} \mathsf{S}_2$*.* ◇

Example 4.5. *Let us take the following two τ!-systems.*



The τ!-system $\mathsf{S}_1 {}^{\mathsf{H}\leftrightarrow\mathsf{K}} \mathsf{S}_2$ is



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

4.1. Compatibility

The notion of compatibility formalises the intuitive requirement that behaviours of the two gateways should match. It is clear that without such a requirement the composition would not behave well. We will see later on that for a successful composition additional technical conditions are needed. Intuitively, two interfaces H and K are *compatible* if:

- each input of one of them has a corresponding (matching) output in the other one.

  Since an input of an interface, say H, is followed in the corresponding gateway by an output towards the gateway for K, this guarantees that each message sent by the gateway for H can be received from the one for K.

- if an output is possible in one of the interfaces, then at least one input (possibly with a message that does not match it) is possible in the other one.

  This ensures that if the gateway for H needs to send messages to S_1, then at least a message can arrive from the gateway for K. However, if the gateway for H can send more than one message, it may be the case that the gateway for K can send only some of them. What happens in this case is that an internal choice in H is delegated to K, which may decide to never choose some of the available options, while always choosing only available options (this follows from the point above).

To define the compatibility relation it is useful to introduce a few auxiliary notions. More precisely, we let $\mathcal{L}_{\mathrm{i/o}} = \{\ ?\mathsf{m}, !\mathsf{m} \mid \mathsf{m} \in \mathcal{M}\ \}$ and borrow from [3] the function $\mathsf{io} : \mathcal{L}_{\mathrm{act}} \to \mathcal{L}_{\mathrm{i/o}}$ defined as follows

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

This function straightforwardly extends to τ-CFSMs. In particular, given $M = \langle \mathcal{S}, q_0, \mathcal{L}_{\text{act}} \cup \{\tau\}, \rightarrow \rangle$ we define $\mathsf{io}(M) = \langle \mathcal{S}, q_0, \mathcal{L}_{\text{i/o}} \cup \{\tau\}, \rightarrow' \rangle$ where

$$\rightarrow' = \{\, q \xrightarrow{\mathsf{io}(\lambda)} q' \mid q \xrightarrow{\lambda} q' \in M, \lambda \in \mathcal{L}_{\text{act}} \,\} \cup \{\, q \xrightarrow{\tau} q' \mid q \xrightarrow{\tau} q' \in M \,\}$$

The function $\mathsf{io}(\_)$ captures the fact that our gateway construction blurs away the identities of the participants that actually sent the messages to their gateway.

Definition 4.6 (Compatibility)**.** *Let M and M' be either two CFSMs or two τ!-CFSMs[4] whose sets of states are respectively $\mathcal{S}$ and $\mathcal{S}'$. A relation $R \subseteq \mathcal{S} \times \mathcal{S}'$ is an* io-matching *of M and M' if whenever $(q, q') \in R$:*

(i) *if $q \xrightarrow{?\mathsf{m}} r \in \mathsf{io}(M)$ then*

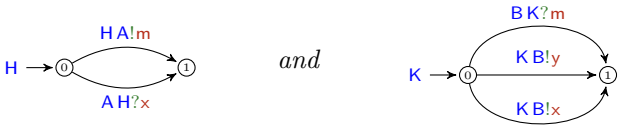
- *either there is $q' \xrightarrow{!\mathsf{m}} r' \in \mathsf{io}(M')$ such that $(r, r') \in R$*
- *or there is $q' \xrightarrow{\tau} \xrightarrow{!\mathsf{m}} r' \in \mathsf{io}(M')$ such that $(r, r') \in R$*

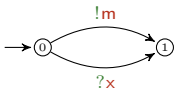
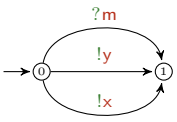
(ii) *if either $q \xrightarrow{!\mathsf{m}} r \in \mathsf{io}(M)$ or $q \xrightarrow{\tau} r \in \mathsf{io}(M)$ then q' has at least one input transition in $\mathsf{io}(M')$.*

An io-correspondence *between M and M' is an io-matching R between M and M' such that R^{-1} is an io-matching between M' and M. We say that two states q and q' are* compatible *(in symbols $q \asymp q'$) if there is an io-correspondence containing (q, q'); M and M' are* compatible *if their initial states are compatible.*

Definition 4.6 transfers the notion of compatibility given in [16] for processes in multiparty sessions. Our compatibility relation is a form of bisimulation with a bias on the input transitions of the interfaces; in fact, we require to match an input transition with an output of the same message while the requirement is weaker for output or τ-transitions. It is worth noticing that such a weaker requirement, namely condition (ii) of Definition 4.6, makes sense only because, when defining $q \asymp q'$, we expect that there exists an io-matching R such that also R^{-1} is so. Such inverse io-matching relation ensures (via condition (i)) that the input transition that should exist matches one of the available outputs. Summarising, the two conditions together ensure that the outputs are a subset of the inputs (see Example 4.7 below). This notion differs from the ones in [6] and in [2, 3] which are defined as bisimulations and do not involve τ-transitions.

Example 4.7 (Compatibility and CFSMs)**.** *The relation $\{(0,0),(0,1)\}$ is an io-correspondence between the states of the CFSMs in Example 4.4, stating that state 0 of M_{H} corresponds to both states 0 and 1 of M_{K}. Therefore, M_{H} and M_{K} are compatible. A more interesting example is given by the following CFSMs:*

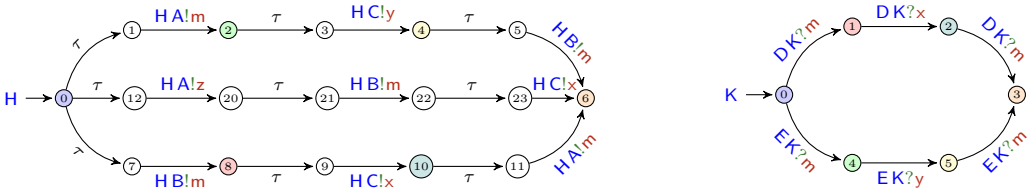


for which we have $\mathsf{io}(M_{\mathsf{H}}) =$  *and* $\mathsf{io}(M_{\mathsf{K}}) =$ 

and it is easy to check that $\{(0,0),(1,1)\}$ is an io-correspondence relating the initial states of M_{H} and M_{K}. Interestingly, the transition $0 \xrightarrow{!\mathsf{y}} 1 \in \mathsf{io}(M_{\mathsf{K}})$ is not matched by an input of $\mathsf{io}(M_{\mathsf{H}})$ since Definition 4.6 requires only that the related state has an input transition. ◇

[4] The definition can be extended in order to take into account τ-CFSMs in general. We chose not to do so for the sake of simplicity, since our results focus on CFSMs and τ!-CFSMs.

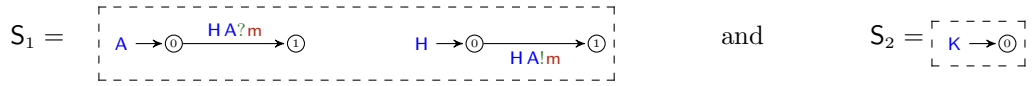
Example 4.8 (Compatibility and τ!-CFSMs)**.** *The τ!-CFSMs M_{H} and M_{K} of Example 4.5 are compatible, in fact, it is easy to check that the relation $\{(0,0),(0,1),(0,2),(1,3)\}$ is an io-correspondence between their states. For a more complex example let us consider the following τ!-CFSMs*



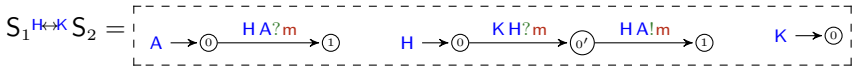
where the states with the same colour are in io-correspondence.

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 input action can find a matching output. The vice versa does not hold, since the output from state 12 in M_{H} is not matched in M_{K}. Such a possibility is in fact allowed by our definition of compatibility. Note that there is no io-correspondence relating state 12 and a state of M_{K}. ◇

Clause (i) in Definition 4.6 ensures that each input of one interface is matched by some output of the other. Clause (ii) requires a milder condition on non-input transitions; in fact, an output or a τ-transition of one interface just force the other interface to be able to execute an input. The latter clause prevents H and K from being compatible in the following systems.



Allowing their compatibility would spoil the deadlock freedom of the composed system:



being its initial configuration immediately stuck. As shown in the next section, the compatibility conditions need to be further strengthened in order to preserve communication properties in the composition operation.

4.2. Composability

We require a stronger condition than compatibility for two systems to be composable. Basically, we need to restrict the non-deterministic behaviour of systems.

Definition 4.9 (Machines' determinism)**.** *An A-local CFSM or τ!-CFSMM is*

1. ?-deterministic *if $p \xrightarrow{\mathsf{X}\,\mathsf{A}?\mathsf{m}} q$ and $p \xrightarrow{\mathsf{Y}\,\mathsf{A}?\mathsf{m}} r \in M$ implies $q = r$;*
2. !-deterministic *if either*
   - *M is a τ!-CFSM and ($p \xrightarrow{\tau}{}^{*} \xrightarrow{\mathsf{A}\,\mathsf{X}!\mathsf{m}} q$ and $p \xrightarrow{\tau}{}^{*} \xrightarrow{\mathsf{A}\,\mathsf{Y}!\mathsf{m}} r \in M$) implies $q = r$; or*
   - *M is a CFSM and ($p \xrightarrow{\mathsf{A}\,\mathsf{X}!\mathsf{m}} q$ and $p \xrightarrow{\mathsf{A}\,\mathsf{Y}!\mathsf{m}} r \in M$) implies $q = r$.*
3. ?!-deterministic *if it is both ?-deterministic and !-deterministic;*

A state $q \in M$ is mixed if it has both at least an outgoing input transition and an outgoing output or τ transition.

Example 4.10. *Machines H and K in Example 4.8 are, respectively, non !-deterministic and non ?-deterministic. In particular, conditions (2) and (1) of Definition 4.9 fail for, respectively, state 0 of H and state 0 of K. For an example of mixed state the reader is referred to Example 7.2 where the initial state of machine K is mixed.*

We now define composability.

Definition 4.11 ((H,K)-composability)**.** *Let S_1 and S_2 be either both communicating systems or both τ!-system. Systems S_1 and S_2 are* (H,K)-composable *if their domains are disjoint, and $\mathsf{H} \in \mathrm{dom}(\mathsf{S}_1)$ and $\mathsf{K} \in \mathrm{dom}(\mathsf{S}_2)$ are two compatible ?!-deterministic machines with no mixed states.*

5. Composed Systems and Their Projections

The general idea underlying most of the proofs of property preservation will be that of showing that if a property does fail for a composed system, so it does for at least one of its composing systems. We introduce a few technical definitions and results to exploit this general proof scheme.

Given a configuration of a composed system, say $\mathsf{S}_1 \,{}^{\mathsf{H}\leftrightarrow\mathsf{K}}\, \mathsf{S}_2$, we can retrieve the configurations of S_1 (resp. S_2) by taking only the states of participants in S_1 (resp. S_2) while mapping fresh states introduced by the gateway construction to states of the original machines. Indeed, we shall prove in Proposition 5.9 that reachable configurations of $\mathsf{S}_1 \,{}^{\mathsf{H}\leftrightarrow\mathsf{K}}\, \mathsf{S}_2$ correspond to reachable configurations of S_1 and of S_2.

Definition 5.1 (Configuration projections)**.** *Let* $\mathsf{S} = \mathsf{S}_1 \,{}^{\mathsf{H}\leftrightarrow\mathsf{K}}\, \mathsf{S}_2$ *be a composed communicating or* τ!*-system and* $s \in \mathcal{R}(\llbracket \mathsf{S} \rrbracket)$ *be a reachable configuration of* S*. The* left-projection of s on S_1 *is the map* ${}^{\mathsf{H}}_{\mathsf{K}}\rfloor s$ *defined on* $\mathrm{dom}(\mathsf{S}_1)$ *by*

$$ {}^{\mathsf{H}}_{\mathsf{K}}\rfloor s : \mathsf{A} \mapsto \begin{cases} q, & \text{if } s(\mathsf{A}) \text{ is a fresh state of a } q\text{-}r \text{ importing segment from } \mathsf{K} \text{ in } \mathsf{gw}(\mathsf{S}_1(\mathsf{H}), \mathsf{K}) \\ r, & \text{if } s(\mathsf{A}) \text{ is a fresh state of a } q\text{-}r \text{ exporting segment to } \mathsf{K} \text{ in } \mathsf{gw}(\mathsf{S}_1(\mathsf{H}), \mathsf{K}) \\ s(\mathsf{A}), & \text{otherwise} \end{cases} $$

The definition of right-projection $s\lfloor^{\mathsf{K}}_{\mathsf{H}}$ *is analogous.*

Proposition 5.2. *Left- and right-projections are well-defined.*

Proof. It is enough to show that ${}^{\mathsf{H}}_{\mathsf{K}}\rfloor\cdot$ and $\cdot\lfloor^{\mathsf{K}}_{\mathsf{H}}$ uniquely assign a state to fresh states because both functions leave unchanged non-fresh states. This follows since, by definition, each state introduced by our gateway constructions belongs to a unique q-r (importing or exporting) segment. □

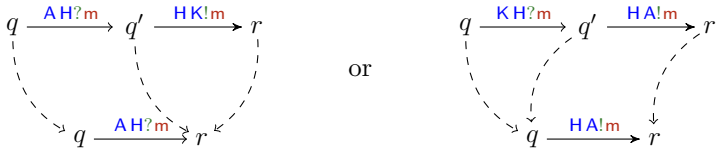
Intuitively, only H is aware of an input from K when H is in the internal state of an importing segment; hence to have a coherent configuration we take the state of H before the input. If instead H is in the internal state of an exporting segment, then other participants in S_1 know that the message has been sent; hence to have a coherent configuration we take the state of H after the output. (A similar intuition applies to $s\lfloor^{\mathsf{K}}_{\mathsf{H}}$.) A pictorial description of Definition 5.1 according to this intuition is provided after Fact 5.3 below.

Fact 5.3. *Let* s *be a configuration of a system* $\mathsf{S}_1 \,{}^{\mathsf{H}\leftrightarrow\mathsf{K}}\, \mathsf{S}_2$*.*

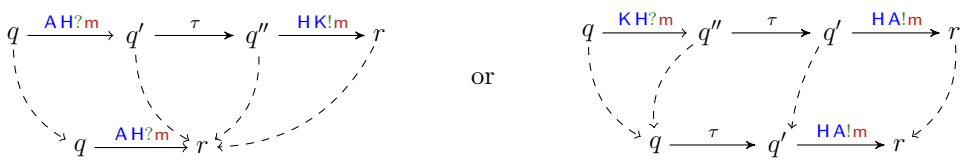
$$ \text{for all } \mathsf{A} \in \mathrm{dom}(\mathsf{S}_1) \setminus \{\mathsf{H}\} \quad {}^{\mathsf{H}}_{\mathsf{K}}\rfloor s(\mathsf{A}) = s(\mathsf{A}) \qquad \text{and} \qquad \text{for all } \mathsf{A} \in \mathrm{dom}(\mathsf{S}_2) \setminus \{\mathsf{K}\} \quad s\lfloor^{\mathsf{K}}_{\mathsf{H}}(\mathsf{A}) = s(\mathsf{A}) $$

This fact does not hold for gateways since, for instance, the local state $s(\mathsf{H})$ can be such that H is engaging in a communication with K; in this case the configuration projection yields a local state in $\mathsf{S}_1(\mathsf{H})$ which depends on the form of the segment $\mathsf{S}(\mathsf{H})$ is on. Notice that in the latter case $\mathsf{S}(\mathsf{H})$ is a gateway while $\mathsf{S}_1(\mathsf{H})$ is the original interface.

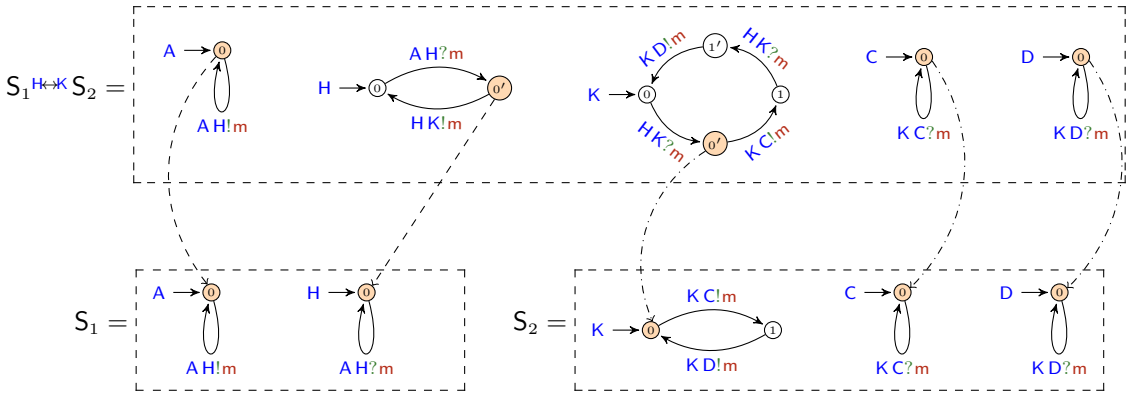
To explain the mapping we appeal to the pictorial representation of Definition 5.1 announced above. Assume first that S_1 and S_2 are communicating systems. By considering transitions in $\mathsf{S}_1(\mathsf{H})$ and their corresponding segment in the gateway, we have



where the dotted edges mark the state assigned by the "left-projection" of s on H. For instance, ${}^{\mathsf{H}}_{\mathsf{K}}\rfloor s(\mathsf{H}) = q$ if $s(\mathsf{H}) = q$ while if $s(\mathsf{H}) = q'$ then ${}^{\mathsf{H}}_{\mathsf{K}}\rfloor s(\mathsf{H}) = r$ on the exporting segment to K and ${}^{\mathsf{H}}_{\mathsf{K}}\rfloor s(\mathsf{H}) = q$ on the importing segment from K. Similarly, assuming that S_1 and S_2 are τ!-systems, we have



Example 5.4. *Consider the communicating systems* S_1, S_2, *and* $\mathsf{S}_1 \,{}^{\mathsf{H}\leftrightarrow\mathsf{K}}\, \mathsf{S}_2$ *of Example 4.4. The next picture shows the projection of a configuration of* $\mathsf{S}_1 \,{}^{\mathsf{H}\leftrightarrow\mathsf{K}}\, \mathsf{S}_2$ *on a configuration of* S_1 *and one of* S_2*, where configurations are represented by filled states. The left-projection* ${}^{\mathsf{H}}_{\mathsf{K}}\rfloor s$ *is the one identified by the dashed lines, whereas* $s\lfloor^{\mathsf{K}}_{\mathsf{H}}$ *is identified through the dash-dotted lines.*



The configuration of $\mathsf{S}_1 \,{}^{\mathsf{H}\leftrightarrow\mathsf{K}}\, \mathsf{S}_2$*, say* s*, is reached from the initial configuration after the following sequence of interactions* $\mathsf{A}{\to}\mathsf{H}\colon \mathsf{m}\cdot\mathsf{H}{\to}\mathsf{K}\colon \mathsf{m}\cdot\mathsf{A}{\to}\mathsf{H}\colon \mathsf{m}$*. Note that such projected configurations are reachable in* S_1 *and* S_2 *respectively.* ◇

Example 5.5. *Let* $s = (2_{\mathsf{A}}, 0_{\mathsf{B}}, 1_{\mathsf{H}}, 3_{\mathsf{K}}, 1_{\mathsf{C}}, 0_{\mathsf{D}}, 0_{\mathsf{E}})$*; and* $\mathsf{S} = \mathsf{S}_1 \,{}^{\mathsf{H}\leftrightarrow\mathsf{K}}\, \mathsf{S}_2$ *be the* $\tau!$*-system of Example 4.5. Then,* $s \in \mathcal{R}(\llbracket \mathsf{S} \rrbracket)$*, namely* s *is reachable in* S*. In fact*

$$
\begin{aligned}
s_0 = (0_{\mathsf{A}}, 0_{\mathsf{B}}, 0_{\mathsf{H}}, 0_{\mathsf{K}}, 0_{\mathsf{C}}, 0_{\mathsf{D}}, 0_{\mathsf{E}}) \quad &\xrightarrow{\tau} \quad s_1 = (1_{\mathsf{A}}, 0_{\mathsf{B}}, 0_{\mathsf{H}}, 0_{\mathsf{K}}, 0_{\mathsf{C}}, 0_{\mathsf{D}}, 0_{\mathsf{E}}) \\
&\xrightarrow{\mathsf{A}\to\mathsf{H}\colon \mathsf{m}} \quad s_2 = (2_{\mathsf{A}}, 0_{\mathsf{B}}, 2_{\mathsf{H}}, 0_{\mathsf{K}}, 0_{\mathsf{C}}, 0_{\mathsf{D}}, 0_{\mathsf{E}}) \\
&\xrightarrow{\tau} \quad s_3 = (2_{\mathsf{A}}, 0_{\mathsf{B}}, 4_{\mathsf{H}}, 0_{\mathsf{K}}, 0_{\mathsf{C}}, 0_{\mathsf{D}}, 0_{\mathsf{E}}) \\
&\xrightarrow{\mathsf{H}\to\mathsf{K}\colon \mathsf{m}} \quad s_4 = (2_{\mathsf{A}}, 0_{\mathsf{B}}, 1_{\mathsf{H}}, 5_{\mathsf{K}}, 0_{\mathsf{C}}, 0_{\mathsf{D}}, 0_{\mathsf{E}}) \\
&\xrightarrow{\tau} \quad s_5 = (2_{\mathsf{A}}, 0_{\mathsf{B}}, 1_{\mathsf{H}}, 1_{\mathsf{K}}, 0_{\mathsf{C}}, 0_{\mathsf{D}}, 0_{\mathsf{E}}) \\
&\xrightarrow{\mathsf{K}\to\mathsf{C}\colon \mathsf{m}} \quad s = (2_{\mathsf{A}}, 0_{\mathsf{B}}, 1_{\mathsf{H}}, 3_{\mathsf{K}}, 1_{\mathsf{C}}, 0_{\mathsf{D}}, 0_{\mathsf{E}})
\end{aligned}
$$

The projections of s *on, respectively,* S_1 *and* S_2 *are* ${}^{\mathsf{H}}_{\mathsf{K}}\rfloor s = (2_{\mathsf{A}}, 0_{\mathsf{B}}, 1_{\mathsf{H}})$ *and* $s\lfloor^{\mathsf{K}}_{\mathsf{H}} = (3_{\mathsf{K}}, 1_{\mathsf{C}}, 0_{\mathsf{D}}, 0_{\mathsf{E}})$ *and they are both reachable in their respective* $\tau!$*-systems:*

$$
\begin{aligned}
{}^{\mathsf{H}}_{\mathsf{K}}\rfloor s_0 = (0_{\mathsf{A}}, 0_{\mathsf{B}}, 0_{\mathsf{H}}) \quad &\xrightarrow{\tau} \quad {}^{\mathsf{H}}_{\mathsf{K}}\rfloor s_1 = (1_{\mathsf{A}}, 0_{\mathsf{B}}, 0_{\mathsf{H}}) \\
&\xrightarrow{\mathsf{A}\to\mathsf{H}\colon \mathsf{m}} \quad {}^{\mathsf{H}}_{\mathsf{K}}\rfloor s_2 = (2_{\mathsf{A}}, 0_{\mathsf{B}}, 1_{\mathsf{H}}) = {}^{\mathsf{H}}_{\mathsf{K}}\rfloor s_3 = {}^{\mathsf{H}}_{\mathsf{K}}\rfloor s_4 = {}^{\mathsf{H}}_{\mathsf{K}}\rfloor s_5 = {}^{\mathsf{H}}_{\mathsf{K}}\rfloor s
\end{aligned}
$$

$$
\begin{aligned}
s_0\lfloor^{\mathsf{K}}_{\mathsf{H}} = s_1\lfloor^{\mathsf{K}}_{\mathsf{H}} = {}^{\mathsf{H}}_{\mathsf{K}}\rfloor s_2 = s_3\lfloor^{\mathsf{K}}_{\mathsf{H}} = s_4\lfloor^{\mathsf{K}}_{\mathsf{H}} = (0_{\mathsf{K}}, 0_{\mathsf{C}}, 0_{\mathsf{D}}, 0_{\mathsf{E}}) \quad &\xrightarrow{\tau} \quad s_5\lfloor^{\mathsf{K}}_{\mathsf{H}} = (1_{\mathsf{K}}, 0_{\mathsf{C}}, 0_{\mathsf{D}}, 0_{\mathsf{E}}) \\
&\xrightarrow{\mathsf{K}\to\mathsf{C}\colon \mathsf{m}} \quad s\lfloor^{\mathsf{K}}_{\mathsf{H}} = (3_{\mathsf{K}}, 1_{\mathsf{C}}, 0_{\mathsf{D}}, 0_{\mathsf{E}})
\end{aligned}
$$

◇

By the above examples it is natural to expect that the projections of configurations that are reachable in a composed system are in turn reachable in the respective component systems. Also, from Example 5.5, it is reasonable to expect that one of the two composing systems has a participant with an enabled transition from one of the projected configurations when a transition is enabled in a configuration of the composed system. This is formalised in Proposition B.1 (cf. Appendix B).

Next definition maps states of gateways to states of the original interfaces, with the aim of showing that the gateways always stay "in sync", mimicking the compatibility relation between the interfaces. To simplify the correspondence, we force the two gateways to change configuration at the same time. Given that the only transition where the two gateways interact is the communication between them, transitions allowing one to change state in an interface correspond to the H-K transition in the gateways. This is clearly visible in the graphical interpretation below.

Definition 5.6 (Gateway embedding)**.** *Let M be an* H*-local CFSM or τ!-CFSM and* $\mathsf{K} \in \mathcal{P}$ *be a participant not occurring in M. The* gateway embedding Γ *maps the states of* $\mathsf{gw}(M, \mathsf{K})$ *to the states of M as follows:*

$$\Gamma_{\mathsf{H},\mathsf{K}}(M, p) = \begin{cases} q, & \text{if } \mathsf{A} \neq \mathsf{K} \text{ and } p \text{ is an internal state of a } q\text{-}r \text{ exporting segment to } \mathsf{K} \text{ in } \mathsf{gw}(M, \mathsf{K}) \\ r, & \text{if } \mathsf{A} \neq \mathsf{K} \text{ and } p \text{ is an internal state of a } q\text{-}r \text{ importing segment from } \mathsf{K} \text{ in } \mathsf{gw}(M, \mathsf{K}) \\ p, & \text{otherwise} \end{cases}$$

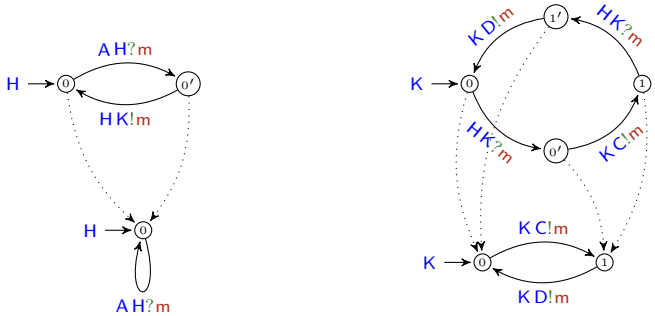
As before we give a pictorial description of the above definition. For CFSMs we have

while for τ!-CFSM we have

By definition of gateway (Definition 4.1) and the above pictures, it immediately descends the following

Fact 5.7. *Function Γ is well-defined.*

Example 5.8. *Let M and M' be the gateways for, respectively, participants* H *and* K *of Example 4.4. The gateway embeddings $\Gamma_{\mathsf{H},\mathsf{K}}(M, \_)$ and $\Gamma_{\mathsf{K},\mathsf{H}}(M', \_)$ map, respectively, the states of M and M' as indicated below by the dotted edges.*

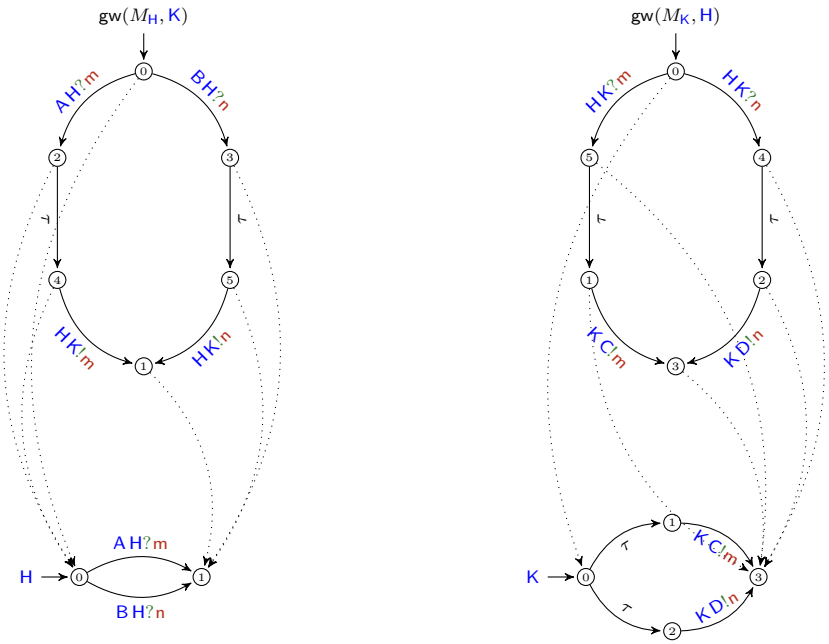


Let M and M' be the gateways for, respectively, participants H *and* K *of Example 4.5. The gateway embeddings $\Gamma_{\mathsf{H},\mathsf{K}}(M, \_)$ and $\Gamma_{\mathsf{K},\mathsf{H}}(M', \_)$ map, respectively, the states of M and M' as indicated below by the dotted*

Notation	Meaning	Introduced in
M_{A}	A-local τ-CFSM	Definition 3.1
$p, q, \ldots$	states of a τ-CFSM	Intro Section 3
$\mathsf{S}, \mathsf{S}', \ldots$	systems of τ-CFSMs	Definition 3.2
$s, s', \ldots$	configurations of a system	Definition 3.3
$[\![\mathsf{S}]\!]$	S's op. semantics (an FSA)	Definition 3.3
$\mathcal{R}([\![\mathsf{S}]\!])$	$[\![\mathsf{S}]\!]$'s reachable configurations	Intro Section 3
$\mathsf{gw}(M, \mathsf{K})$	gateway towards K of the τ-CFSM M	Definition 4.1
$\mathsf{S}_1 {}^{\mathsf{H}}\!\!\times\!\!{}_{\mathsf{K}}\, \mathsf{S}_2$	composition via H and K of S_1 and S_2	Definition 4.3
$q \asymp q'$	compatible states	Definition 4.6
$\binom{\mathsf{H}}{\mathsf{K}}\!\rfloor s$ $(s\lfloor\binom{\mathsf{K}}{\mathsf{H}})$	left(right)-projection of s	Definition 5.1
$\Gamma_{\mathsf{H},\mathsf{K}}(M, p)$	gateway embedding of the state p of M	Definition 5.6

Table 2: Main notations

edges.



◇

The notion of gateway embedding is similar to configuration projection (but for considering a single state instead of a whole configuration); a main change is that, when $\mathsf{gw}(M_{\mathsf{H}}, \mathsf{K})$ receives a message from its own system S_1 going to some fresh state q'', the gateway embedding $\Gamma_{\mathsf{H},\mathsf{K}}$ maps q'' to the previous state since S_2, and K in particular, are not yet aware of the transition. Instead 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. Thus, gateway embeddings are designed to establish a correspondence with the other system as shown by the next proposition (whose proof is in Appendix A). Roughly, we show that by left/right-projecting a reachable configuration of a composed system, we get reachable configurations in the component systems. Moreover, these latter are compatible. For convenience Table 2 recaps the main notations introduced so far and used below.

Proposition 5.9. *Let* S_1 *and* S_2 *be two* (H,K)*-composable communicating systems or* $\tau!$*-systems. If* $s \in \mathcal{R}([\![\mathsf{S}_1 {}^{\mathsf{H}\leftrightarrow\mathsf{K}} \mathsf{S}_2]\!])$ *then* ${}^{\mathsf{H}}_{\mathsf{K}}\rfloor s \in \mathcal{R}([\![\mathsf{S}_1]\!])$*,* $s\lfloor^{\mathsf{K}}_{\mathsf{H}} \in \mathcal{R}([\![\mathsf{S}_2]\!])$*, and* $\Gamma_{\mathsf{H},\mathsf{K}}(\mathsf{S}_1(\mathsf{H}), s(\mathsf{H}))$ *and* $\Gamma_{\mathsf{K},\mathsf{H}}(\mathsf{S}_2(\mathsf{K}), s(\mathsf{K}))$ *are compatible (cf. Definition 4.6).*

6. What does Composability Buy Us?

Composition via gateways does ensure the preservation of deadlock freedom assuming composability of $\tau!$-systems. This follows since 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 will imply 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.

This fact in turn will be obtained using the following proposition, whose proof will be provided in Appendix B. Roughly, it states that if a projection of a system composed of deadlock-free subsystems can move then also the composed system moves.

Proposition 6.1 (Weak lifting of $\tau!$-systems)**.** *Let* S_1 *and* S_2 *be two* (H,K)*-composable and deadlock-free* $\tau!$*-systems and let* $\mathsf{S} = \mathsf{S}_1 {}^{\mathsf{H}\leftrightarrow\mathsf{K}} \mathsf{S}_2$*. For all* $s \in \mathcal{R}([\![\mathsf{S}]\!])$ *if* ${}^{\mathsf{H}}_{\mathsf{K}}\rfloor s \rightarrow$ *or* $s\lfloor^{\mathsf{K}}_{\mathsf{H}} \rightarrow$ *then* $s \rightarrow$*.*

Theorem 6.2 (Deadlock freedom preservation for $\tau!$-systems)**.** *Let* S_1 *and* S_2 *be two* (H,K)*-composable and deadlock-free* $\tau!$*-systems. The composed* $\tau!$*-system* $\mathsf{S}_1 {}^{\mathsf{H}\leftrightarrow\mathsf{K}} \mathsf{S}_2$ *is deadlock-free.*

Proof. We show that if the composed $\tau!$-system $\mathsf{S}_1 {}^{\mathsf{H}\leftrightarrow\mathsf{K}} \mathsf{S}_2$ reaches a deadlock configuration s then at least one of ${}^{\mathsf{H}}_{\mathsf{K}}\rfloor s$ and $s\lfloor^{\mathsf{K}}_{\mathsf{H}}$ is a deadlock. This descends immediately from the fact that if $s \in \mathcal{R}([\![\mathsf{S}_1 {}^{\mathsf{H}\leftrightarrow\mathsf{K}} \mathsf{S}_2]\!])$ enables a participant in $\mathsf{S}_1 {}^{\mathsf{H}\leftrightarrow\mathsf{K}} \mathsf{S}_2$ then ${}^{\mathsf{H}}_{\mathsf{K}}\rfloor s$ or $s\lfloor^{\mathsf{K}}_{\mathsf{H}}$ enable one of their participants (cf. Proposition B.1) once we show that

$$s \not\rightarrow \quad \Longrightarrow \quad {}^{\mathsf{H}}_{\mathsf{K}}\rfloor s \not\rightarrow \quad \text{and} \quad s\lfloor^{\mathsf{K}}_{\mathsf{H}} \not\rightarrow$$

This can be obtained as the contrapositive of Proposition 6.1 □

Example 6.3. *We can infer deadlock-freedom of the* $\tau!$*-system* $\mathsf{S} = \mathsf{S}_1 {}^{\mathsf{H}\leftrightarrow\mathsf{K}} \mathsf{S}_2$ *of Example 4.5 by the result above, since* S_1 *and* S_2 *are* (H,K)*-composable and deadlock-free.*

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

Theorem 6.4 (Strong lock freedom preservation for $\tau!$-systems)**.** *Let* S_1 *and* S_2 *be two* (H,K)*-composable and strongly lock free* $\tau!$*-systems. The composed* $\tau!$*-system* $\mathsf{S} = \mathsf{S}_1 {}^{\mathsf{H}\leftrightarrow\mathsf{K}} \mathsf{S}_2$ *is strongly lock free.*

Proof. Let X be a participant enabled in a reachable configuration $s \in \mathcal{R}([\![\mathsf{S}]\!])$ of S. Without loss of generality, we can assume $\mathsf{X} \in \mathsf{ptp}(\mathsf{S}_1)$. By contradiction, let ψ be a maximal fair run starting from s not involving X and let $s(\mathsf{X}) \xrightarrow{\lambda} q$ be a transition of $\mathsf{S}(\mathsf{X})$. Necessarily, $\mathsf{H},\mathsf{K} \in \mathsf{ptp}(\lambda)$ otherwise by strong lock freedom of S_1 and fairness there would be transitions on ψ involving the participants of λ contrary to our hypothesis. Notice that if λ is an output from H towards S_1 then the transformation to gateway turns an internal choice of H (due to τ transitions) to a choice of K due to the H-K interactions. However, in both cases strong lock freedom applies to all the possible options of the choice, hence we can appeal to the strong lock freedom of S_1 for each of them.

Hence, either of the following cases is possible:

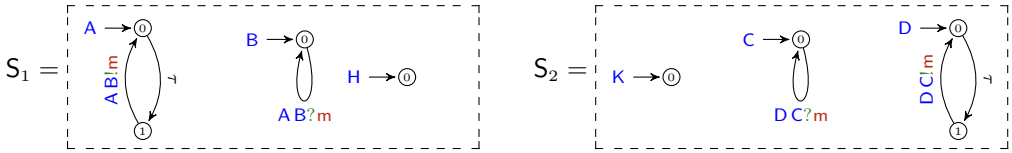
- $\lambda = \mathsf{H} \rightarrow \mathsf{K} \colon \mathsf{m}$. The gateway $\mathsf{gw}(M_{\mathsf{K}}, \mathsf{H})$ has an input enabled from H in $\mathsf{S}(\mathsf{K})$; by Proposition 5.9 each input from $\mathsf{S}(\mathsf{K})$ is matched by an output of $\mathsf{S}(\mathsf{H})$ (possibly after a τ-transition) of $\mathsf{gw}(M_{\mathsf{H}}, \mathsf{K})$ and, by fairness of ψ, there should be a transition on ψ involving H.

- $\lambda = \mathsf{K} \rightarrow \mathsf{H} \colon \mathsf{m}$. By Proposition 5.9 each input from $\mathsf{S}(\mathsf{H})$ is matched by an output of $\mathsf{S}(\mathsf{K})$ (possibly after a τ-transition) of $\mathsf{gw}(M_{\mathsf{K}}, \mathsf{H})$ and, by fairness of ψ, there should be a transition on ψ involving H.

Both cases contradict the assumption that H is not involved on ψ and we get the thesis. □

Notice that dropping the restriction to fair runs in the definition of strong lock freedom (Definition 3.9) would falsify the above theorem, as shown in the simple following example.

Example 6.5 (Unfair runs spoil strong lock freedom preservation)**.** *The following τ!-systems are trivially* (H,K)*-composable and strongly lock free.*

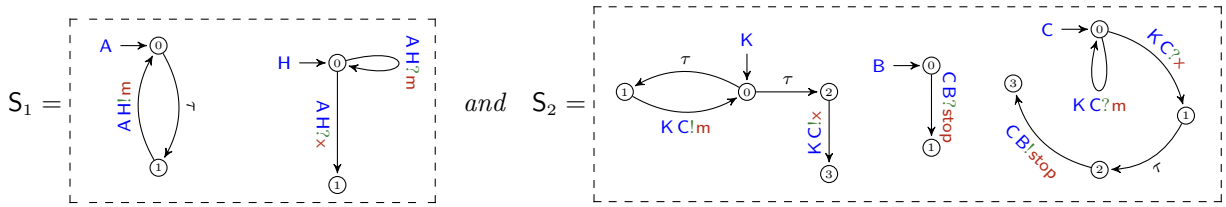


The composition of the above τ!-systems is obtained by simply putting them side by side. In such a case, if we did not restrict ourselves to fair runs in the definition of strong lock freedom, the composition would be non strongly lock free. In fact the initial configuration would be a weak lock for A *by considering the (unfair) maximal infinite run where* C *and* D *keep on exchanging the message* m*.*

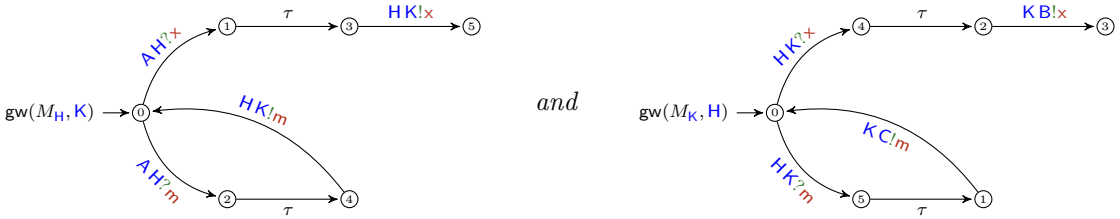
The present counterexample immediately adapts to communicating systems by simply taking out the τ transitions. ◇

Turning now our attention to lock freedom, it is possible to show, by the following example, that composability is too weak a property to preserve lock freedom in the case of asymmetric synchronous interactions.

Example 6.6 (Composability does not preserve lock-freedom for τ!-systems)**.** *Let us consider the following τ!-systems.*



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



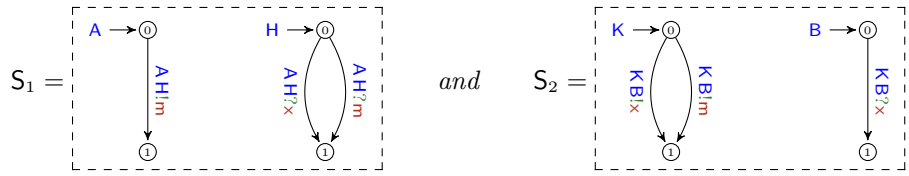
The initial configuration of the composed τ!-system, say s_0, is a lock for B *since the only outgoing transition from $0 \in M_{\mathsf{B}}$ could be fired only in case the transition* C B!stop *is enabled. However, this is impossible since* A *always sends message* m*, which is forwarded by* $\mathsf{gw}(M_{\mathsf{H}},\mathsf{K})$ *towards* S_2*; hence, in* $[\![\mathsf{S}_1 \overset{\mathsf{H}}{+} \mathsf{S}_2]\!]$ *only the following run is present:*

$$
\begin{array}{rcl}
s_0 & \xrightarrow{\tau} & (1_{\mathsf{A}}, 0_{\mathsf{gw}(M_{\mathsf{H}},\mathsf{K})}, 0_{\mathsf{gw}(M_{\mathsf{K}},\mathsf{H})}, 0_{\mathsf{B}}, 0_{\mathsf{C}}) \\
 & \xrightarrow{\mathsf{A}\to\mathsf{H}:\,\mathsf{m}} & (0_{\mathsf{A}}, 2_{\mathsf{gw}(M_{\mathsf{H}},\mathsf{K})}, 0_{\mathsf{gw}(M_{\mathsf{K}},\mathsf{H})}, 0_{\mathsf{B}}, 0_{\mathsf{C}}) \\
 & \xrightarrow{\tau} & (0_{\mathsf{A}}, 4_{\mathsf{gw}(M_{\mathsf{H}},\mathsf{K})}, 0_{\mathsf{gw}(M_{\mathsf{K}},\mathsf{H})}, 0_{\mathsf{B}}, 0_{\mathsf{C}}) \\
 & \xrightarrow{\mathsf{H}\to\mathsf{K}:\,\mathsf{m}} & (0_{\mathsf{A}}, 0_{\mathsf{gw}(M_{\mathsf{H}},\mathsf{K})}, 5_{\mathsf{gw}(M_{\mathsf{K}},\mathsf{H})}, 0_{\mathsf{B}}, 0_{\mathsf{C}}) \\
 & \xrightarrow{\tau} & (0_{\mathsf{A}}, 0_{\mathsf{gw}(M_{\mathsf{H}},\mathsf{K})}, 1_{\mathsf{gw}(M_{\mathsf{K}},\mathsf{H})}, 0_{\mathsf{B}}, 0_{\mathsf{C}}) \\
 & \xrightarrow{\mathsf{K}\to\mathsf{C}:\,\mathsf{m}} & s_0 \\
 & \cdots &
\end{array}
$$

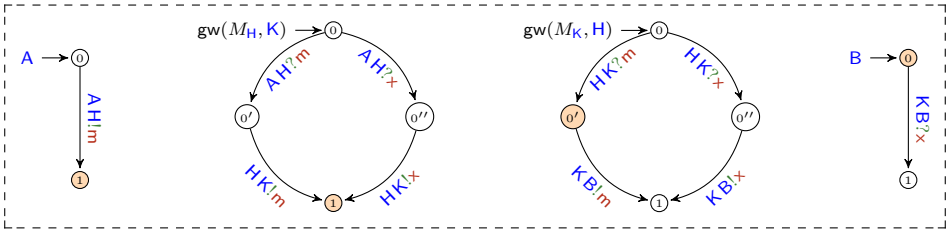
where the above shown transitions (which do not involve B*) are perpetually executed, leaving no chance to* B *of ever firing the only outgoing transition from its initial state.* ◇

Somehow surprisingly, in the symmetric case none of the properties of Definition 3.9 is preserved by composition if only composability is required.

Example 6.7 (Symmetric synchronisation spoils communication properties)**.** *Take the following communicating systems:*



Clearly, S_1 *and* S_2 *are* (H,K)*-composable and strongly lock-free (hence also lock- and deadlock-free). However, their composition* $\mathsf{S}_1 \overset{\mathsf{H}\leftrightarrow\mathsf{K}}{} \mathsf{S}_2$ *reaches the following configuration (identified by the filled states):*

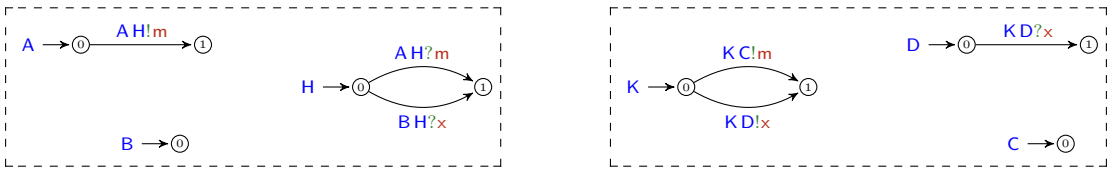


which is a deadlock (and hence the composition is neither deadlock-, nor lock-, nor strongly lock-free). In fact such configuration is reached when the gateway for K *receives* m*, it tries to synchronise with participant* B *on message* m *while* B *is waiting only for* x*. For* S_2 *in isolation, this is not a deadlock, since* B *and* K *synchronise on* x *under the symmetric semantics.* ◇

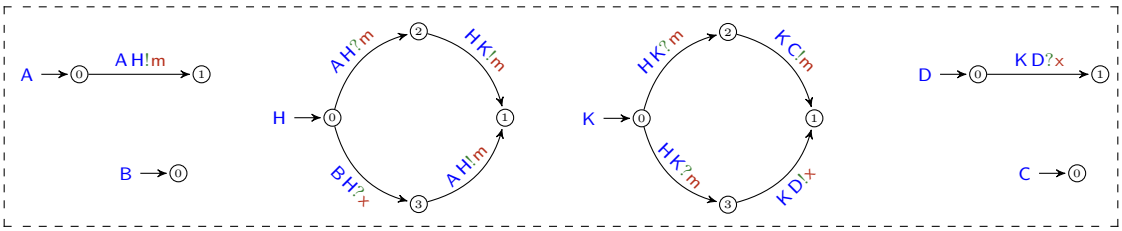
This example shows that deadlock freedom would not be preserved even by strengthening the notion of compatibility by disregarding our bias on inputs, like in [6, Def. 3.6]. The example above, instead, is not a counterexample in an asynchronous setting. Indeed, S_2 could deadlock due to the fact that K could send m without synchronising with B. Likewise, Example 6.7 does not yield a counterexample in our asymmetric setting. In fact, despite synchronicity, 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 K B!m.

The following examples show that the compositionality issue is due to distributed choices, hence it also emerges in slightly different contexts than in Example 6.7.

Example 6.8 (Distributed choices may spoil communication properties)**.** *The composition of the deadlock-free communicating systems below*



is the following system

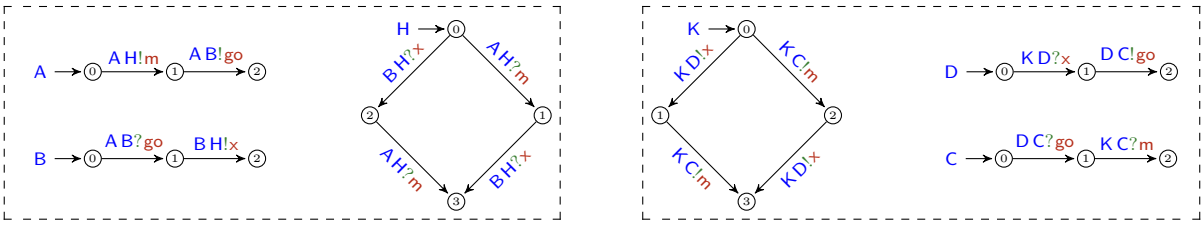


The system above can reach a deadlock, similarly to what happens in Example 6.7. In particular, the interactions A→H: m·H→K: m *lead to the configuration*

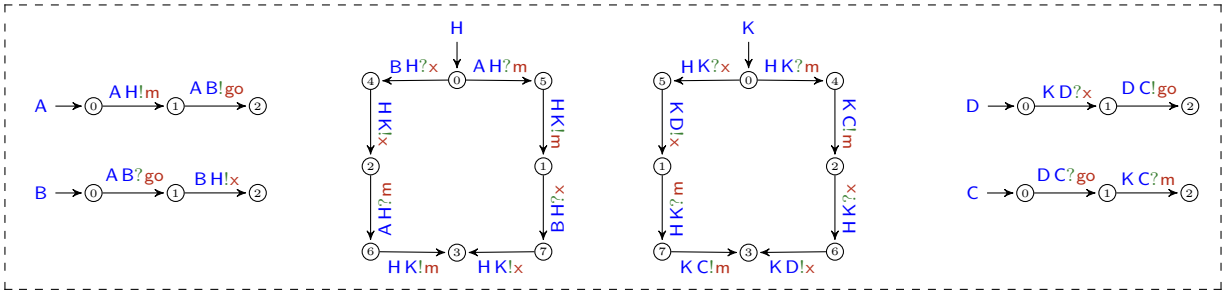
$$(1_{\mathsf{A}}, 0_{\mathsf{B}}, 1_{\mathsf{H}}, 2_{\mathsf{K}}, 0_{\mathsf{D}}, 0_{\mathsf{C}})$$

which is a deadlock since the system is stuck but K *and* D *have not terminated: the former is willing to send* m *while the latter to receive* x*. However, in Example 6.7 the choice is resolved by a single participant (i.e.,* A *in* S_1 *and* B *in* S_2*) while here the choices involve communications with different participants (i.e.,* A *and* B *on the left and* C *and* D *on the right).* ◇

Example 6.9 (Concurrency may spoil communication properties)**.** *Take the communicating systems below*



Their composition is as follows:



Similarly to Example 6.7, the composed communicating system deadlocks while the two communicating systems in isolation do not. In particular, after the interactions A→H: m·H→K: m·A→B: go·B→H: x *the system reaches the configuration*

$$(2_{\mathsf{A}}, 2_{\mathsf{B}}, 1_{\mathsf{H}}, 4_{\mathsf{K}}, 0_{\mathsf{D}}, 0_{\mathsf{C}})$$

which is a deadlock. Indeed, A *and* B *force an order in the exchange with the gateway while* C *and* D *expect the opposite order. Hence also choices embedded in commuting diamonds are problematic.* ◇

The examples above clarify that any property in the symmetric synchronous setting and lock-freedom in the asymmetric case can be preserved only under stricter conditions on interfaces than composability. In Section 8 we shall show that, by recurring to interface participants not containing choices, we can recover the missing preservation properties.

7. No Composability, No Party

Violating one of the conditions of Definition 4.11 may compromise the validity of Theorem 6.2 and Theorem 6.4. We first show that dropping the compatibility condition can falsify both theorems.

Example 7.1 (Compatibility is needed)**.** *Let us consider the following* τ!*-systems:*

These τ!-systems trivially enjoy all the communication properties of Definition 3.9. 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 $\mathsf{S}_1 \not\leftrightarrow \mathsf{S}_2$, the following transitions are possible in $[\![\mathsf{S}_1 \not\leftrightarrow \mathsf{S}_2]\!]$

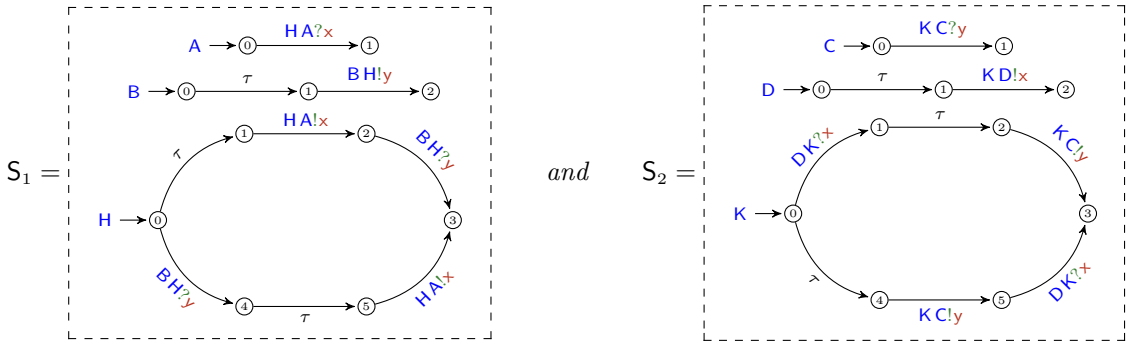
$$(0_\mathsf{A}, 0_\mathsf{H}, 0_\mathsf{K}, 0_\mathsf{C}) \xrightarrow{\tau} (0_\mathsf{A}, 0_\mathsf{H}, 0_\mathsf{K}, 1_\mathsf{C}) \xrightarrow{\mathsf{C} \longrightarrow \mathsf{K} : \mathsf{y}} (0_\mathsf{A}, 0_\mathsf{H}, 2_\mathsf{K}, 2_\mathsf{C}) \xrightarrow{\tau} (0_\mathsf{A}, 0_\mathsf{H},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

where $(0_\mathsf{A}, 0_\mathsf{H}, 3_\mathsf{K}, 2_\mathsf{C})$ is a deadlock configuration (hence also a lock and a weak lock) for $\mathsf{S}_1 \not\leftrightarrow \mathsf{S}_2$ since K wishes to send message y to H, which is instead waiting for message x.

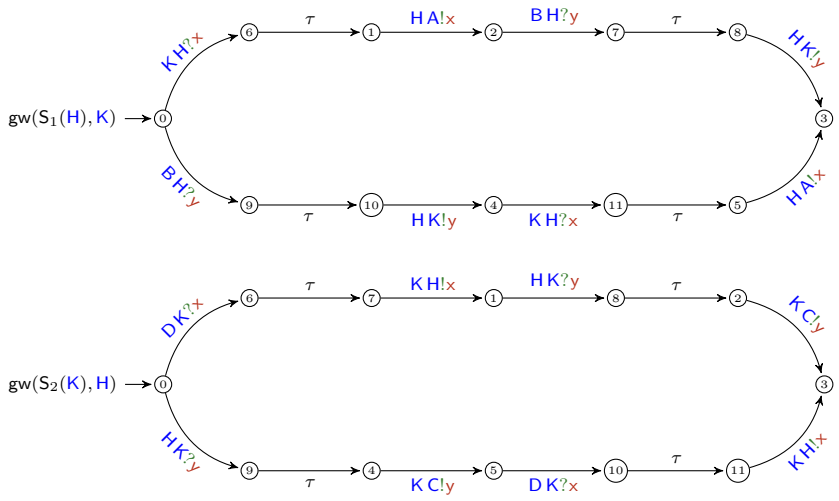
By removing τ-transitions in the above machines we immediately get a counterexample for the case of communicating systems in the symmetric synchronous case. ◇

The following example casts in our setting an example given in [3] for the asynchronous semantics. This example illustrates that Theorem 6.2 and Theorem 6.4 fail in presence of mixed states.

Example 7.2 (Mixed-states spoil communication properties)**.** *Let*



The two τ!-systems are strongly lock free (and hence necessarily lock- and deadlock-free). Moreover, notice that the initial states are mixed and that H and K are compatible. The gateways we obtain are

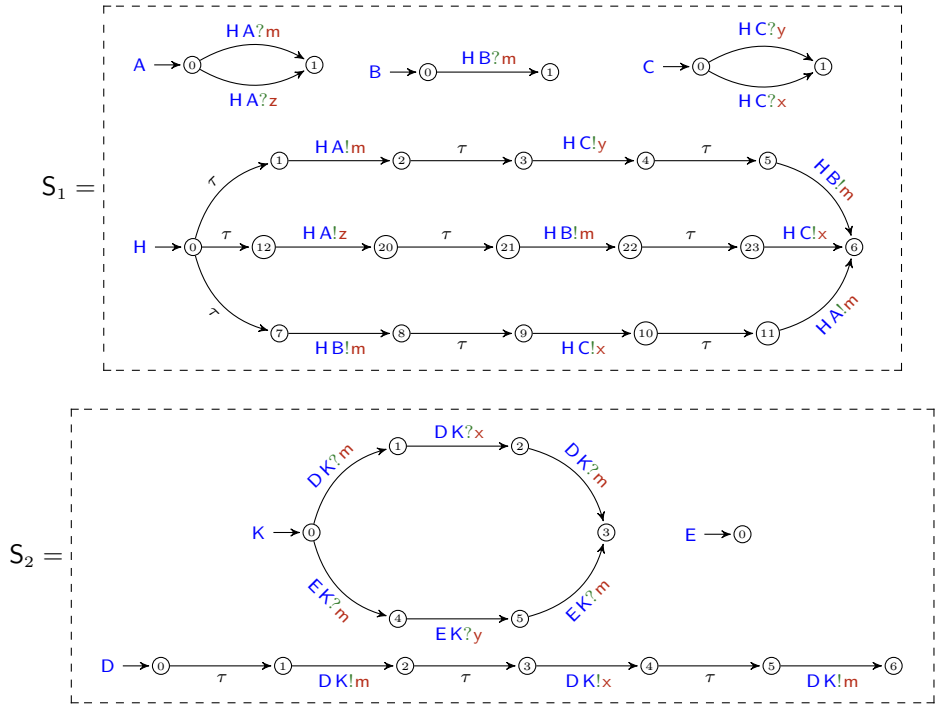


The composed τ!-system $\mathsf{S}_1 \overset{\mathsf{H}\leftrightarrow\mathsf{K}}{\bowtie} \mathsf{S}_2$ deadlocks when $\mathsf{gw}(\mathsf{S}_1(\mathsf{H}), \mathsf{K})$ receives from B while $\mathsf{gw}(\mathsf{S}_2(\mathsf{K}), \mathsf{H})$ receives from D since both gateways reach an output state (respectively states 10 and 7). Recall that this necessarily implies such a τ!-system to be neither lock-, strong lock- nor deadlock-free.

By removing τ-transitions in the above machines we immediately get also a counterexample for the case of communicating systems in the symmetric synchronous case. ◇

Example 7.4 below shows that !?-nondeterminism also spoils Theorems 6.2 and 6.4 by taking the composition of the two τ!-systems in Example 7.3, which features a non !?-deterministic deadlock-free τ!-system.

Example 7.3 (Two deadlock free τ!-systems)**.** *The following two τ!-systems whose participants H and K are from Example 4.8,*



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. Since

- *K is !?-nondeterministic since from its initial state K can reach two different states (1 and 4) by receiving message m and*
- *E cannot synchronise since M_E does not have any transition.*

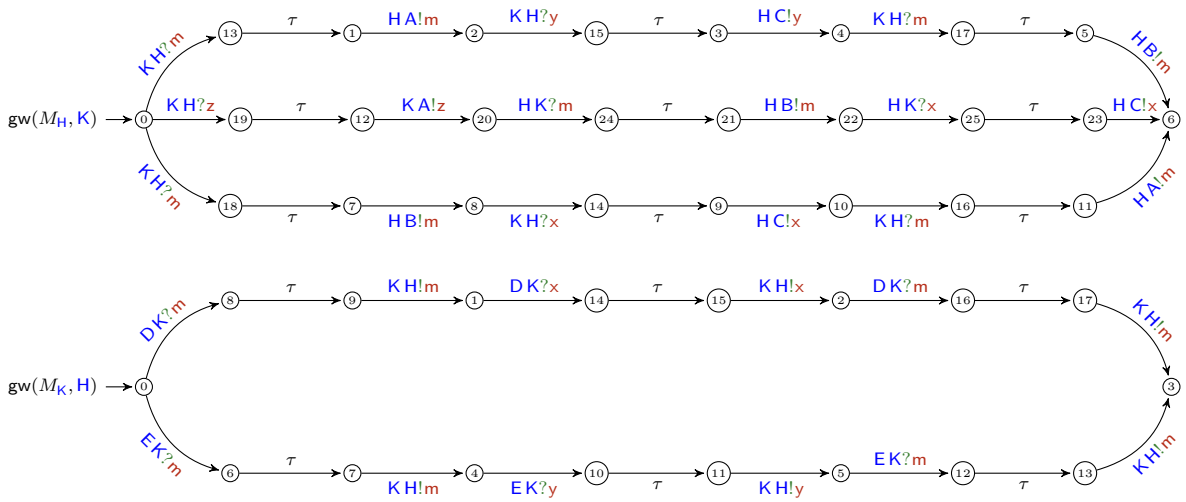
the only possible transitions of S_2 must involve K and D only. We therefore have that

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

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 τ!-systems S_1 and S_2 in Example 7.3 can reach a deadlock.

Example 7.4 (!?-Nondeterminism spoils deadlock freedom)**.** *The* τ!*-CFSMs* H *and* K *in Example 7.3 are compatible as seen in Example 4.8. Hence, we can build the composed* τ!*-system* $\mathsf{S}_1 \,{}^{\mathsf{H}\not\leftrightarrow\mathsf{K}}\, \mathsf{S}_2$ *through the gateways*



Now, from the initial configuration of $\mathsf{S}_1 \,{}^{\mathsf{H}\not\leftrightarrow\mathsf{K}}\, \mathsf{S}_2$*, say* s_0*, 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 *gateway* $\mathsf{gw}(M_{\mathsf{H}}, \mathsf{K})$ *can move either to state* 18 *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* τ!*-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* $\mathsf{S}_1 \,{}^{\mathsf{H}\not\leftrightarrow\mathsf{K}}\, \mathsf{S}_2$*, 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. Intuitively, the upper path of* $\mathsf{gw}(M_{\mathsf{H}}, \mathsf{K})$ *should be matched by the lower path of* $\mathsf{gw}(M_{\mathsf{K}}, \mathsf{H})$*. However, since the exchanged message is the same, the choice is actually non-deterministic, hence the wrong combination may happen, and this leads to a deadlock. By removing* τ*-transitions in the above automata we immediately get a counterexample for the case of communicating systems in the symmetric synchronous case.*

Observe that by Proposition 3.10 the reached deadlock configuration is also a (weak) lock configuration. ◇

8. Raise the Bet with Sequentiality and Win the Pot

As mentioned at the end of Section 6, it is possible to show that composition by gateways does preserve all the communication properties if we recur to interface participants not containing choices. We dub such participants *sequential*.

Definition 8.1 (Sequential τ-CFSM)**.** *A τ-CFSM is* sequential *if each of its states has at most one outgoing transition. A participant A of a τ-system S is sequential if so is $\mathsf{S}(\mathsf{A})$.*

By adding the requirement of sequentiality of gateways to composability we manage to show that lock-freedom is preserved in the asymmetric case, as well as all the communication properties in the symmetric case. We note that sequentiality implies absence of mixed states and ?!-determinism, while the converse does not hold. Moreover, the problems of Examples 6.6 and 6.7 cannot happen in case we restrict to sequential gateways.

The following proposition (see Appendix C for the proof) is instrumental for the proof of lock-freedom preservation because it allows us to restructure a run by having τ steps just before the corresponding output.

Proposition 8.2 (τ delaying)**.** *Given a τ!-system S, for all $s \in \mathcal{R}(\llbracket \mathsf{S} \rrbracket)$, if $s \xrightarrow{\lambda} s'$ and $\lambda \neq \tau$ then there is $\hat{s} \in \mathcal{R}(\llbracket \mathsf{S} \rrbracket)$ such that $\hat{s} \xrightarrow{\tau} s \xrightarrow{\lambda} s'$ and the sender of λ is involved in $\hat{s} \xrightarrow{\tau} s$.*

The key point in the proof of lock-freedom preservation is the construction of a run in $\mathsf{S}_1 \,{}^{\mathsf{H}\!\!\rightarrow\!\!\leftarrow\!\!\mathsf{K}}\, \mathsf{S}_2$ out of a run in either S_1 or S_2. This is obtained in Proposition 8.3 below (see Appendix C for the proof) by exploiting sequentiality and lock-freedom of S_1 and S_2 in order to guarantee the construction when the run "crosses" the gateways.

Proposition 8.3 (Transitions lifting)**.** *Let S_1 and S_2 be two (H,K)-composable and lock-free communicating or τ!-systems with H and K sequential and let $\mathsf{S} = \mathsf{S}_1 \,{}^{\mathsf{H}\!\!\rightarrow\!\!\leftarrow\!\!\mathsf{K}}\, \mathsf{S}_2$. For all $s \in \mathcal{R}(\llbracket \mathsf{S} \rrbracket)$ and labels λ, if ${}^{\mathsf{H}}_{\mathsf{K}}\rfloor s \xrightarrow{\lambda} \bar{s}$ in $\llbracket \mathsf{S}_1 \rrbracket$ then there is a configuration $\hat{s} \in \mathcal{R}(\llbracket \mathsf{S} \rrbracket, s)$ such that $\hat{s} \xrightarrow{\lambda} s'$ and ${}^{\mathsf{H}}_{\mathsf{K}}\rfloor s' = \bar{s}$.*
The same holds for the right-projection.

We can now prove lock-freedom preservation by contradiction.

Theorem 8.4 (Lock-freedom preservation)**.** *Let S_1 and S_2 be two (H,K)-composable and lock-free communicating or τ!-systems with H and K sequential. Then the composed system $\mathsf{S}_1 \,{}^{\mathsf{H}\!\!\rightarrow\!\!\leftarrow\!\!\mathsf{K}}\, \mathsf{S}_2$ is lock-free.*

Proof. By contradiction, let us assume $\mathsf{S} = \mathsf{S}_1 \,{}^{\mathsf{H}\!\!\rightarrow\!\!\leftarrow\!\!\mathsf{K}}\, \mathsf{S}_2$ not to be lock-free. Then there is a configuration $s \in \mathcal{R}(\llbracket \mathsf{S} \rrbracket)$ and a participant $\mathsf{X} \in \mathsf{S}$ not involved in any run from s. Without loss of generality, we can assume $\mathsf{X} \in \mathsf{S}_1$. We have ${}^{\mathsf{H}}_{\mathsf{K}}\rfloor s \in \mathcal{R}(\llbracket \mathsf{S}_1 \rrbracket)$ by Proposition 5.9 and, by lock-freedom of S_1, ${}^{\mathsf{H}}_{\mathsf{K}}\rfloor s$ cannot be a lock of S_1 for X. Hence, there exists a run

$$ {}^{\mathsf{H}}_{\mathsf{K}}\rfloor s \xrightarrow{\lambda_1} \hat{s}_1 \cdots \hat{s}_{n-1} \xrightarrow{\lambda_n} \hat{s}_n \tag{10} $$

of S_1 with X involved in λ_n. This induces a run from s in S involving X by repeated applications of Proposition 8.3 (if S_1 and S_2 are τ!-systems, then by Proposition 8.2 we can safely assume run (10) to be such that each output transition is immediately preceded by the corresponding τ-transition; namely the τ-transition involving the sender of the output action). So s reaches a configuration with a run involving X, which contradicts our assumption. □

The fact that a composed τ!-system $\mathsf{S} = \mathsf{S}_1 \,{}^{\mathsf{H}\!\!\rightarrow\!\!\leftarrow\!\!\mathsf{K}}\, \mathsf{S}_2$ enabling a participant induces one of the component τ!-systems S_1 or S_2 to enable one of their participants may, at a first glance, be considered trivial. However, proving this property is not completely straightforward since interfaces are transformed into gateways by our composition operation.

Proposition 8.5 (Weak lifting of communicating systems)**.** *Let S_1 and S_2 be two (H,K)-composable and deadlock-free communicating systems with H and K sequential and let $\mathsf{S} = \mathsf{S}_1 \,{}^{\mathsf{H}\!\!\rightarrow\!\!\leftarrow\!\!\mathsf{K}}\, \mathsf{S}_2$. For all $s \in \mathcal{R}(\llbracket \mathsf{S} \rrbracket)$, if ${}^{\mathsf{H}}_{\mathsf{K}}\rfloor s \rightarrow$ or $s \lfloor^{\mathsf{K}}_{\mathsf{H}} \rightarrow$ then $s \rightarrow$.*

Proposition 8.5 together with Proposition 8.6 below are used in the proofs of our last two theorems. More precisely, given $\mathsf{S} = \mathsf{S}_1 \,{}^{\mathsf{H}}\!\!\not\!\!\ast^{\mathsf{K}}\, \mathsf{S}_2$ and one of its reachable configurations, say s, Proposition 8.5 shows that a transition from the left/right projection of s can be lifted to a transition from s itself. Proposition 8.6 does roughly the opposite, but in a stronger way: a transition from s can be put in correspondence with a transition from its left/right projection. Besides, the left/right projection of the configuration reached from s is the one reached from the left/right projection of s. This means that transitions and projections partly commute.

Proposition 8.6 (Reflecting interactions on sub-systems)**.** *Let* $\mathsf{S} = \mathsf{S}_1 \,{}^{\mathsf{H}}\!\!\not\!\!\ast^{\mathsf{K}}\, \mathsf{S}_2$ *be the composition of* S_1 *and* S_2*, two* (H,K)*-composable communicating systems or* $\tau!$*-systems. For all* $s \in \mathcal{R}([\![\mathsf{S}]\!])$ *and* $s \xrightarrow{\lambda} s'$*,*

i) *if* $\lambda = \tau$ *then* $({}^{\mathsf{H}}_{\mathsf{K}}\rfloor s \xrightarrow{\tau} {}^{\mathsf{H}}_{\mathsf{K}}\rfloor s'$ *or* ${}^{\mathsf{H}}_{\mathsf{K}}\rfloor s = {}^{\mathsf{H}}_{\mathsf{K}}\rfloor s')$

ii) *if* $\lambda \neq \tau$ *then* ${}^{\mathsf{H}}_{\mathsf{K}}\rfloor s \xrightarrow{\lambda} {}^{\mathsf{H}}_{\mathsf{K}}\rfloor s'$*, or* ${}^{\mathsf{H}}_{\mathsf{K}}\rfloor s \xrightarrow{\tau} \xrightarrow{\lambda} {}^{\mathsf{H}}_{\mathsf{K}}\rfloor s'$*, or else* ${}^{\mathsf{H}}_{\mathsf{K}}\rfloor s = {}^{\mathsf{H}}_{\mathsf{K}}\rfloor s'$*.*

The same holds for the $\_\lfloor^{\mathsf{K}}_{\mathsf{H}}$ *projection.*

Theorem 8.7 (Deadlock freedom preservation for communicating systems)**.** *Let* S_1 *and* S_2 *be two* (H,K)*-composable and deadlock-free communicating systems, where* H *and* K *are sequential. Then the composed communicating system* $\mathsf{S}_1 \,{}^{\mathsf{H}}\!\!\not\!\!\ast^{\mathsf{K}}\, \mathsf{S}_2$ *is deadlock-free.*

Proof. We show that if the composed communicating system $\mathsf{S} = \mathsf{S}_1 \,{}^{\mathsf{H}}\!\!\not\!\!\ast^{\mathsf{K}}\, \mathsf{S}_2$ reaches a deadlock configuration s then at least one of ${}^{\mathsf{H}}_{\mathsf{K}}\rfloor s$ and $s\lfloor^{\mathsf{K}}_{\mathsf{H}}$ is a deadlock. This descends immediately from the fact that if $s \in \mathcal{R}([\![\mathsf{S}]\!])$ enables a participant in S then ${}^{\mathsf{H}}_{\mathsf{K}}\rfloor s$ or $s\lfloor^{\mathsf{K}}_{\mathsf{H}}$ enable one of their participants (cf. Proposition B.1) once we show that

$$s \not\rightarrow \quad \Longrightarrow \quad {}^{\mathsf{H}}_{\mathsf{K}}\rfloor s \not\rightarrow \quad \text{and} \quad s\lfloor^{\mathsf{K}}_{\mathsf{H}} \not\rightarrow$$

This can be obtained as the contrapositive of Proposition 8.5. □

Theorem 8.8 (Strong lock freedom preservation for communicating systems)**.** *Let* S_1 *and* S_2 *be two* (H,K)*-composable and strongly lock free communicating systems, with* H *and* K *sequential. Then the composed communicating system* $\mathsf{S}_1 \,{}^{\mathsf{H}}\!\!\not\!\!\ast^{\mathsf{K}}\, \mathsf{S}_2$ *is strongly lock free.*

Proof. By contradiction, let us assume $\mathsf{S} = \mathsf{S}_1 \,{}^{\mathsf{H}}\!\!\not\!\!\ast^{\mathsf{K}}\, \mathsf{S}_2$ not to be strongly lock free. This means that there are a reachable configuration s, a participant X enabled in s, and a maximal run ξ from s not involving X. Without loss of generality, we can assume X to be a participant of S_1. By repeated application of Proposition 8.6, there exists a run of S_1

$${}^{\mathsf{H}}_{\mathsf{K}}\rfloor s \xrightarrow{\lambda_1} s_1 \cdots \xrightarrow{\lambda_n} s_n \cdots \tag{11}$$

with X not involved in ξ. If the run (11) is infinite then it is necessarily maximal; otherwise, let $\hat{s}$ be the last configuration in (11). Then it must be $\hat{s} \not\rightarrow$ otherwise, by Proposition 8.5, ξ would not be maximal. In both cases we get a contradiction of the hypothesis that S_1 is strongly lock free since we found a maximal run (11) in S_1 not involving X. □

9. Conclusion, Related and Future Work

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, that we call gateways, connecting the systems, as investigated initially in [2, 3] for an asynchronous interaction model, namely the one of Communicating Finite State Machines (CFSMs) [4]. Relevant communication properties were shown to be preserved by the "composition-by-gateway" mechanism. The investigation on such composition method and the properties preserved by it was shifted in [6] towards synchronous

symmetric interactions. In the present paper we pushed such an investigation forward in two directions: (i) by considering other communication properties besides deadlock-freedom (the only one taken into account in [6]); (ii) by considering also an asymmetric synchronous semantics of systems, which breaks the symmetry between senders and receivers. In fact, our asymmetric semantics decouples communication from choice resolution as in standard synchronous semantics of communicating systems (and other models). This is (roughly) accomplished by simply inserting internal silent actions before outputs actions, so yielding what we dub τ!-CFSMs. We then adapted the gateway composition mechanism defined in [2, 3] to our asymmetric semantics and gave conditions for the preservation of relevant communication properties under this notion of composition. We also extended the preservation results in [6] for the symmetric synchronous semantics.

While in the symmetric case, composability is too weak a requirement for preserving any of the communication properties we considered, it is enough to ensure deadlock- and lock-freedom preservation in the asymmetric case. Notably, sequentiality is needed here for lock-freedom preservation, in both the symmetric and asymmetric cases. Interestingly, deadlock- and strong lock-freedom preservation in the synchronous asymmetric case do not require sequentiality of gateways, like in the symmetric case.

Intuitively, τ!-CFSMs can be looked at as the communicating automata counterpart of the τC contracts described in [17, 9]. Imposing the no-mixed state condition (roughly, absence of choices between input and output actions) on our τ!-CFSMs, turns them into the communicating automata counterpart of the processes (contracts) called "session behaviours"[5] in, e.g., [18, 19, 10]. These processes are in turn the process counterpart of (binary) session types [20]. As shown in the paper (and in [3] for asynchronous communications), the absence of mixed states is needed in order to get the preservation of properties under composition. An interesting future work is to check if our compatibility notion falls in the family of compliance relation characterised as fixed points in [21, 22].

In *contract automata* [23, 24] 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 [23, 24].

While the path of investigation above is quite homogeneous, the different analyses present some methodological differences. For instance, for deadlock-freedom in the case of symmetric synchronous interactions, [6] considers also another form of composition, dubbed *semi-direct*, where one single gateway (interacting with both the composed communicating systems) is used. It will be also worth investigating conditions different from sequentiality guaranteeing properties preservation under composition. An interesting one is the weak form of liveness (dubbed !-liveness) taken into account in [6], guaranteeing deadlock-freedom preservation with symmetric synchronous interactions. Any output action of a !-live interface can be fired in some possible evolution of the communicating system.

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] R. Milner, A Calculus of Communicating Systems, Vol. 92 of LNCS, Springer, Berlin, 1980. doi:10.1007/3-540-10235-3.

[2] F. Barbanera, U. de'Liguoro, R. Hennicker, Global types for open systems, in: M. Bartoletti, S. Knight (Eds.), ICE, Vol. 279 of EPTCS, 2018, pp. 4–20. doi:10.4204/EPTCS.279.4.

[3] F. Barbanera, U. de'Liguoro, R. Hennicker, Connecting open systems of communicating finite state machines, JLAMP 109 (2019). doi:10.1016/j.jlamp.2019.07.004.

[4] D. Brand, P. Zafiropulo, On communicating finite-state machines, J. ACM 30 (2) (1983) 323–342. doi:10.1145/322374.322380.

[5] F. Barbanera, I. Lanese, E. Tuosto, On composing communicating systems, in: C. Aubert, C. Di Giusto, L. Safina, A. Scalas (Eds.), Proceedings 15th Interaction and Concurrency Experience, ICE 2022, Lucca, Italy, 17th June 2022, Vol. 365 of EPTCS, 2022, pp. 53–68. doi:10.4204/EPTCS.365.4.
URL https://doi.org/10.4204/EPTCS.365.4

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

[6] F. Barbanera, I. Lanese, E. Tuosto, Composing communicating systems, synchronously, in: T. Margaria, B. Steffen (Eds.), ISoLA 2020, Vol. 12476 of LNCS, Springer, 2020, pp. 39–59. doi:10.1007/978-3-030-61362-4_3.
[7] R. De Nicola, M. Hennessy, CCS without τ's, in: TAPSOFT, Vol.1, Vol. 249 of LNCS, Springer, 1987, pp. 138–152.
[8] H. Hüttel, et al., Foundations of session types and behavioural contracts, ACM Comput. Surv. 49 (1) (2016) 3:1–3:36. doi:10.1145/2873052.
[9] M. Bartoletti, T. Cimoli, R. Zunino, Compliance in behavioural contracts: A brief survey, in: C. Bodei, G. L. Ferrari, C. Priami (Eds.), Programming Languages with Applications to Biology and Security - Essays Dedicated to Pierpaolo Degano on the Occasion of His 65th Birthday, Vol. 9465 of Lecture Notes in Computer Science, Springer, 2015, pp. 103–121. doi:10.1007/978-3-319-25527-9_9.
[10] M. Bartoletti, A. Scalas, R. Zunino, A semantic deconstruction of session types, in: P. Baldan, D. Gorla (Eds.), CONCUR 2014, Vol. 8704 of LNCS, Springer, 2014, pp. 402–418. doi:10.1007/978-3-662-44584-6_28.
[11] L. Padovani, Contract-based discovery of web services modulo simple orchestrators, Theoretical Computer Science 411 (2010) 3328–3347. doi:10.1016/j.tcs.2010.05.002.
[12] N. Francez, C. A. R. Hoare, D. J. Lehmann, W. P. de Roever, Semantics of nondeterminism, concurrency, and communication, J. Comput. Syst. Sci. 19 (3) (1979) 290–308.
[13] G. Cécé, A. Finkel, Verification of programs with half-duplex communication, I&C 202 (2) (2005) 166–190. doi:10.1016/j.ic.2005.05.006.
[14] J. Lange, E. Tuosto, N. Yoshida, From communicating machines to graphical choreographies, in: S. K. Rajamani, D. Walker (Eds.), POPL, ACM, 2015, pp. 221–232. doi:10.1145/2676726.2676964.
[15] E. Tuosto, R. Guanciale, Semantics of global view of choreographies, JLAMP 95 (2018) 17–40. doi:10.1016/j.jlamp.2017.11.002.
[16] F. Barbanera, M. Dezani-Ciancaglini, I. Lanese, E. Tuosto, Composition and decomposition of multiparty sessions, JLAMP 119 (2021) 100620. doi:10.1016/j.jlamp.2020.100620.
[17] M. Bartoletti, M. Murgia, A. Scalas, R. Zunino, Verifiable abstractions for contract-oriented systems, J. Log. Algebraic Methods Program. 86 (1) (2017) 159–207. doi:10.1016/j.jlamp.2015.10.005.
URL https://doi.org/10.1016/j.jlamp.2015.10.005
[18] G. Bernardi, M. Hennessy, Modelling session types using contracts, in: Proceedings of the 27th Annual ACM Symposium on Applied Computing, SAC '12, ACM, New York, NY, USA, 2012, pp. 1941–1946. doi:10.1145/2231936.2232097.
[19] F. Barbanera, U. de'Liguoro, Sub-behaviour relations for session-based client/server systems, MSCS 25 (6) (2015) 1339–1381. doi:10.1017/S096012951400005X.
[20] K. Honda, V. T. Vasconcelos, M. Kubo, Language primitives and type disciplines for structured communication-based programming, in: C. Hankin (Ed.), ESOP, Vol. 1381 of LNCS, Springer, 1998, pp. 22–138. doi:10.1007/BFb0053567.
[21] M. Murgia, A fixed-points based framework for compliance of behavioural contracts, J. Log. Algebraic Methods Program. 120 (2021) 100641. doi:10.1016/j.jlamp.2021.100641.
URL https://doi.org/10.1016/j.jlamp.2021.100641
[22] M. Murgia, A note on compliance relations and fixed points, in: M. Bartoletti, L. Henrio, A. Mavridou, A. Scalas (Eds.), Proceedings 12th Interaction and Concurrency Experience, ICE 2019, Copenhagen, Denmark, 20-21 June 2019, Vol. 304 of EPTCS, 2019, pp. 38–47. doi:10.4204/EPTCS.304.3.
URL https://doi.org/10.4204/EPTCS.304.3
[23] D. Basile, P. Degano, G. L. Ferrari, E. Tuosto, Playing with our CAT and communication-centric applications, in: E. Albert, I. Lanese (Eds.), FORTE, Vol. 9688 of LNCS, Springer, 2016, pp. 62–73. doi:10.1007/978-3-319-39570-8_5.
[24] D. Basile, M. H. ter Beek, R. Pugliese, Synthesis of orchestrations and choreographies: Bridging the gap between supervisory control and coordination of services, LMCS 16 (2) (2020). doi:10.23638/LMCS-16(2:9)2020.

Appendix A. Proofs for Section 5 (Composed Systems and Their Projections)

Proposition 5.9. *Let* S_1 *and* S_2 *be two* (H,K)*-composable communicating systems or* τ!*-systems. If* $s \in \mathcal{R}(\llbracket \mathsf{S}_1 \leftrightarrows \mathsf{S}_2 \rrbracket)$ *then* ${}^{\mathsf{H}}_{\mathsf{K}}\rfloor s \in \mathcal{R}(\llbracket \mathsf{S}_1 \rrbracket)$, $s\lfloor^{\mathsf{K}}_{\mathsf{H}} \in \mathcal{R}(\llbracket \mathsf{S}_2 \rrbracket)$, *and* $\Gamma_{\mathsf{H},\mathsf{K}}(\mathsf{S}_1(\mathsf{H}), s(\mathsf{H}))$ *and* $\Gamma_{\mathsf{K},\mathsf{H}}(\mathsf{S}_2(\mathsf{K}), s(\mathsf{K}))$ *are compatible (cf. Definition 4.6).*

Proof. We consider the case of τ!-CFSMs. The proof for CFSMs can be obtained by following the same schema, disregarding τ transitions.

Let s_0 be the initial configuration of $\mathsf{S} = \mathsf{S}_1 \leftrightarrows \mathsf{S}_2$ and

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

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

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

- ${}^{\mathsf{H}}_{\mathsf{K}}\rfloor s \in \mathcal{R}(\llbracket \mathsf{S}_1 \rrbracket)$ and $s\lfloor^{\mathsf{K}}_{\mathsf{H}} \in \mathcal{R}(\llbracket \mathsf{S}_2 \rrbracket)$ because left- and right-projections are the initial configurations of S_1 and S_2 respectively, and
- $\Gamma_{\mathsf{H},\mathsf{K}}(\mathsf{S}_1(\mathsf{H}), s(\mathsf{H}))$ and $\Gamma_{\mathsf{K},\mathsf{H}}(\mathsf{S}_2(\mathsf{K}), s(\mathsf{K}))$ 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 involved participants are in S_1. Then, by construction (cf. Definition 5.1), $s\lfloor^{\mathsf{K}}_{\mathsf{H}} = s_{n-1}\lfloor^{\mathsf{K}}_{\mathsf{H}}$ and by the inductive hypothesis $s_{n-1}\lfloor^{\mathsf{K}}_{\mathsf{H}} \in \mathcal{R}(\llbracket \mathsf{S}_2 \rrbracket)$. Moreover, ${}^{\mathsf{H}}_{\mathsf{K}}\rfloor s$ equals ${}^{\mathsf{H}}_{\mathsf{K}}\rfloor s_{n-1}$ but for the local states of the participants involved in λ_n. By the induction hypothesis, ${}^{\mathsf{H}}_{\mathsf{K}}\rfloor s_{n-1} \in \mathcal{R}(\llbracket \mathsf{S}_1 \rrbracket)$; moreover there is a transition from ${}^{\mathsf{H}}_{\mathsf{K}}\rfloor s_{n-1}$ to ${}^{\mathsf{H}}_{\mathsf{K}}\rfloor s_n$ by the semantics of τ!-systems (cf. Definition 3.3). Hence ${}^{\mathsf{H}}_{\mathsf{K}}\rfloor s_n \in \mathcal{R}(\llbracket \mathsf{S}_1 \rrbracket)$. Besides,

$$\begin{array}{rcll} \Gamma_{\mathsf{H},\mathsf{K}}(\mathsf{S}_1(\mathsf{H}), s(\mathsf{H})) & = & \Gamma_{\mathsf{H},\mathsf{K}}(\mathsf{S}_1(\mathsf{H}), s_{n-1}(\mathsf{H})) & \text{because } s(\mathsf{H}) = s_{n-1}(\mathsf{H}) \\ & \asymp & \Gamma_{\mathsf{K},\mathsf{H}}(\mathsf{S}_2(\mathsf{K}), s_{n-1}(\mathsf{K})) & \text{(by the induction hypothesis)} \\ & = & \Gamma_{\mathsf{K},\mathsf{H}}(\mathsf{S}_2(\mathsf{K}), s(\mathsf{K})) & \text{because } s(\mathsf{K}) = s_{n-1}(\mathsf{K}) \end{array}$$

We now consider the case when 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 4.1) only one of the following two cases is possible for the transitions in $\mathsf{gw}(\mathsf{S}_1(\mathsf{H}), \mathsf{K})$:

$$q \xrightarrow{\mathsf{A}\,\mathsf{H}?\mathsf{m}} s_{n-1}(\mathsf{H}) \xrightarrow{\tau} p \xrightarrow{\mathsf{H}\,\mathsf{K}!\mathsf{m}} r \tag{A.2}$$

$$p \xrightarrow{\mathsf{K}\,\mathsf{H}?\mathsf{m}} s_{n-1}(\mathsf{H}) \xrightarrow{\tau} q \xrightarrow{\mathsf{H}\,\mathsf{A}!\mathsf{m}} r \tag{A.3}$$

for some participant $\mathsf{A} \neq \mathsf{H}$ of S_1 and states p, q, r of $\mathsf{gw}(\mathsf{S}_1(\mathsf{H}), \mathsf{K})$. Cases (A.3) and (A.2) respectively correspond to have the transitions

$$p \xrightarrow{\tau} q \xrightarrow{\mathsf{H}\,\mathsf{A}!\mathsf{m}} r \qquad \text{and} \qquad q \xrightarrow{\mathsf{A}\,\mathsf{H}?\mathsf{m}} r$$

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

We consider the case (A.2) first since it is simpler than the other. Since K is not involved in λ_n, we have that $s\lfloor^{\mathsf{K}}_{\mathsf{H}} = s_{n-1}\lfloor^{\mathsf{K}}_{\mathsf{H}}$ and hence $s\lfloor^{\mathsf{K}}_{\mathsf{H}} \in \mathcal{R}(\llbracket \mathsf{S}_2 \rrbracket)$. Moreover, s and s_{n-1} do coincide, but for the local state of H and $s(\mathsf{H}) = p$. Now, by definition of projection, ${}^{\mathsf{H}}_{\mathsf{K}}\rfloor s_{n-1}(\mathsf{H}) = r = {}^{\mathsf{H}}_{\mathsf{K}}\rfloor s(\mathsf{H})$ and by the induction hypothesis ${}^{\mathsf{H}}_{\mathsf{K}}\rfloor s_{n-1} \in \mathcal{R}(\llbracket \mathsf{S}_1 \rrbracket)$. So, we get ${}^{\mathsf{H}}_{\mathsf{K}}\rfloor s \in \mathcal{R}(\llbracket \mathsf{S}_1 \rrbracket)$.

Let us show compatibility. First, observe that $\Gamma_{\mathsf{H},\mathsf{K}}(\mathsf{S}_1(\mathsf{H}), s_{n-1}(\mathsf{H})) \asymp \Gamma_{\mathsf{K},\mathsf{H}}(\mathsf{S}_2(\mathsf{K}), s_{n-1}(\mathsf{K}))$ (by the induction hypothesis). The thesis then follows by $s\lfloor^{\mathsf{K}}_{\mathsf{H}} = s_{n-1}\lfloor^{\mathsf{K}}_{\mathsf{H}}$ (as observed previously), and by $\Gamma_{\mathsf{H},\mathsf{K}}(\mathsf{S}_1(\mathsf{H}), s_{n-1}(\mathsf{H})) = \Gamma_{\mathsf{H},\mathsf{K}}(\mathsf{S}_1(\mathsf{H}), s(\mathsf{H}))$ (by definition of Γ).

Let us now consider case (A.3). By the induction hypothesis, $^{\mathsf{H}}_{\mathsf{K}}\rfloor s_{n-1} \in \mathcal{R}(\llbracket \mathsf{S}_1 \rrbracket)$ and $s_{n-1}\lfloor^{\mathsf{K}}_{\mathsf{H}} \in \mathcal{R}(\llbracket \mathsf{S}_2 \rrbracket)$. Since K is not involved in λ_n, we have that $s\lfloor^{\mathsf{K}}_{\mathsf{H}} = s_{n-1}\lfloor^{\mathsf{K}}_{\mathsf{H}}$ and hence $s\lfloor^{\mathsf{K}}_{\mathsf{H}} \in \mathcal{R}(\llbracket \mathsf{S}_2 \rrbracket)$. Moreover, s and s_{n-1} do coincide, but for the local state of H in s_{n-1}, which becomes $s(\mathsf{H}) = q$ in s. By definition of projections (Definition 5.1), it immediately follows that $^{\mathsf{H}}_{\mathsf{K}}\rfloor s \in \mathcal{R}(\llbracket \mathsf{S}_1 \rrbracket)$.

To show the compatibility of the states, we first notice that the last transition in the run (A.1) must be preceded by a transition, say the i-th one, where the τ!-CFSMs $\mathsf{gw}(\mathsf{S}_1(\mathsf{H}), \mathsf{K})$ and $\mathsf{gw}(\mathsf{S}_2(\mathsf{K}), \mathsf{H})$ have exchanged message m. We have hence that $s_i(\mathsf{H}) = p$ and $\mathsf{gw}(\mathsf{S}_2(\mathsf{K}), \mathsf{H})$ has a transition $s_i(\mathsf{K}) \xrightarrow{\mathsf{K}\,\mathsf{H}!\mathsf{m}} s_{i+1}(\mathsf{K})$. By definition of gateway and Γ, there is a participant $\mathsf{C} \in \mathsf{S}_2$ such that $\Gamma_{\mathsf{K},\mathsf{H}}(\mathsf{S}_2(\mathsf{K}), s_i(\mathsf{K})) \xrightarrow{\mathsf{C}\,\mathsf{K}?\mathsf{m}} s_{i+1}(\mathsf{K}) = \Gamma_{\mathsf{K},\mathsf{H}}(\mathsf{S}_2(\mathsf{K}), s_{i+1}(\mathsf{K}))$. Besides, $p \asymp \Gamma_{\mathsf{K},\mathsf{H}}(\mathsf{S}_2(\mathsf{K}), s_i(\mathsf{K}))$ since

$$\begin{aligned} &\Gamma_{\mathsf{H},\mathsf{K}}(\mathsf{S}_1(\mathsf{H}), s_i(\mathsf{H})) \asymp \Gamma_{\mathsf{K},\mathsf{H}}(\mathsf{S}_2(\mathsf{K}), s_i(\mathsf{K})) && \text{by the induction hypothesis} \\ \text{and } &\Gamma_{\mathsf{H},\mathsf{K}}(\mathsf{S}_1(\mathsf{H}), s_i(\mathsf{H})) = \Gamma_{\mathsf{H},\mathsf{K}}(\mathsf{S}_1(\mathsf{H}), p) = p && \text{by definition of } \Gamma. \end{aligned}$$

So, from $p \xrightarrow{\tau} q \xrightarrow{\mathsf{H}\,\mathsf{A}!\mathsf{m}} r$ and $\Gamma_{\mathsf{K},\mathsf{H}}(\mathsf{S}_2(\mathsf{K}), s_i(\mathsf{K})) \xrightarrow{\mathsf{C}\,\mathsf{H}?\mathsf{m}} \Gamma_{\mathsf{K},\mathsf{H}}(\mathsf{S}_2(\mathsf{K}), s_{i+1}(\mathsf{K}))$ and the fact that r is unique by ?!-determinism, $r \asymp \Gamma_{\mathsf{K},\mathsf{H}}(\mathsf{S}_2(\mathsf{K}), s_{i+1}(\mathsf{K}))$ by compatibility. Also, since $\Gamma_{\mathsf{H},\mathsf{K}}(\mathsf{S}_1(\mathsf{H}), s(\mathsf{H})) = r$ (by definition of Γ), we have

$$\Gamma_{\mathsf{H},\mathsf{K}}(\mathsf{S}_1(\mathsf{H}), s(\mathsf{H})) \asymp \Gamma_{\mathsf{K},\mathsf{H}}(\mathsf{S}_2(\mathsf{K}), s_{i+1}(\mathsf{K}))$$

We now observe that either $s(\mathsf{K}) = s_{i+1}(\mathsf{K})$ or $s(\mathsf{K})$ is an internal state that $\mathsf{gw}(\mathsf{S}_2(\mathsf{K}), \mathsf{H})$ reaches starting from $s_{i+1}(\mathsf{K})$ after having received a message from a participant of S_2 between the $(i+1)$-th and the last interactions in (A.1). The thesis is immediate in the first case, while in the second case, the thesis follows by observing $\Gamma_{\mathsf{H},\mathsf{K}}(\mathsf{S}_2(\mathsf{H}), s_{i+1}(\mathsf{K})) = \Gamma_{\mathsf{K},\mathsf{H}}(\mathsf{S}_2(\mathsf{K}), s(\mathsf{K}))$ (by definition of Γ, cf. Definition 5.6).

$\boxed{\lambda_n = \mathsf{H} \to \mathsf{X} \colon \mathsf{m}}$ By construction, either $\mathsf{X} = \mathsf{K}$ or $\mathsf{X} \neq \mathsf{H}$ is a participant of S_1.

Case $\mathsf{X} = \mathsf{K}$. In the former case, there are participants $\mathsf{A} \in \mathsf{S}_1$ and $\mathsf{B} \in \mathsf{S}_2$ such that $\mathsf{gw}(\mathsf{S}_1(\mathsf{H}), \mathsf{K})$ and $\mathsf{gw}(\mathsf{S}_2(\mathsf{K}), \mathsf{H})$ respectively have the sequences of transitions

$$p \xrightarrow{\mathsf{A}\,\mathsf{H}?\mathsf{m}} \xrightarrow{\tau} s_{n-1}(\mathsf{H}) \xrightarrow{\mathsf{H}\,\mathsf{K}!\mathsf{m}} s(\mathsf{H}) \qquad \text{and} \qquad s_{n-1}(\mathsf{K}) \xrightarrow{\mathsf{H}\,\mathsf{K}?\mathsf{m}} s(\mathsf{K}) \xrightarrow{\tau} \xrightarrow{\mathsf{K}\,\mathsf{B}!\mathsf{m}} r$$

respectively obtained through our gateway construction on the following sequences of $\mathsf{S}_1(\mathsf{H})$ and $\mathsf{S}_2(\mathsf{K})$

$$p \xrightarrow{\mathsf{A}\,\mathsf{H}?\mathsf{m}} \xrightarrow{\tau} s(\mathsf{H}) \qquad \text{and} \qquad s_{n-1}(\mathsf{K}) \xrightarrow{\mathsf{K}\,\mathsf{B}!\mathsf{m}} r$$

We also notice that $^{\mathsf{H}}_{\mathsf{K}}\rfloor s$ equals $^{\mathsf{H}}_{\mathsf{K}}\rfloor s_{n-1}$ but for the local states of H and K. Now, by the induction hypothesis we have $^{\mathsf{H}}_{\mathsf{K}}\rfloor s_{n-1} \in \mathcal{R}(\llbracket \mathsf{S}_1 \rrbracket)$ and $s_{n-1}\lfloor^{\mathsf{K}}_{\mathsf{H}} \in \mathcal{R}(\llbracket \mathsf{S}_2 \rrbracket)$. Also, by definition of projection (cf. Definition 5.1) $^{\mathsf{H}}_{\mathsf{K}}\rfloor s_{n-1}(\mathsf{H}) = {}^{\mathsf{H}}_{\mathsf{K}}\rfloor s(\mathsf{H}) = s(\mathsf{H})$ and $s_{n-1}\lfloor^{\mathsf{K}}_{\mathsf{H}}(\mathsf{K}) = s\lfloor^{\mathsf{K}}_{\mathsf{H}}(\mathsf{K})$. This implies that $^{\mathsf{H}}_{\mathsf{K}}\rfloor s \in \mathcal{R}(\llbracket \mathsf{S}_1 \rrbracket)$ and $s\lfloor^{\mathsf{K}}_{\mathsf{H}} \in \mathcal{R}(\llbracket \mathsf{S}_2 \rrbracket)$.

To show the compatibility part, we first observe that

$$p = \Gamma_{\mathsf{H},\mathsf{K}}(\mathsf{S}_1(\mathsf{H}), s_{n-1}(\mathsf{H})) \qquad \text{and} \qquad \Gamma_{\mathsf{K},\mathsf{H}}(\mathsf{S}_2(\mathsf{K}), s_{n-1}(\mathsf{K})) = s_{n-1}(\mathsf{K})$$

by definition of Γ (cf. Definition 5.6). Therefore, $p \asymp s_{n-1}(\mathsf{K})$ by the inductive hypothesis. Necessarily, we have $s(\mathsf{H}) \asymp r$ by ?!-determinism and definition of compatibility. Finally, by definition of Γ (cf. Definition 5.6), $\Gamma_{\mathsf{H},\mathsf{K}}(\mathsf{S}_1(\mathsf{H}), s(\mathsf{H})) = s(\mathsf{H})$ and $r = \Gamma_{\mathsf{K},\mathsf{H}}(\mathsf{S}_2(\mathsf{K}), s(\mathsf{K}))$ we get the thesis.

Case $\mathsf{X} \in \mathsf{S}_1$. We have $\mathsf{X} \neq \mathsf{H}$ by construction (cf. Definition 3.3) hence the inductive hypothesis immediately entails $s\lfloor^{\mathsf{K}}_{\mathsf{H}} \in \mathcal{R}(\llbracket \mathsf{S}_2 \rrbracket)$ since $s\lfloor^{\mathsf{K}}_{\mathsf{H}} = s_{n-1}\lfloor^{\mathsf{K}}_{\mathsf{H}}$ because K is not involved in λ_n. We now show the reachability of the left–projection.

By construction (cf. Definition 4.1), there is a sequence $p \xrightarrow{\tau} s_{n-1}(\mathsf{H}) \xrightarrow{\mathsf{H}\,\mathsf{X}!\mathsf{m}} s(\mathsf{H})$ in $\mathsf{S}_1(\mathsf{H})$ which yields the importing sequence

$$p \xrightarrow{\mathsf{K}\,\mathsf{H}?\mathsf{m}} q \xrightarrow{\tau} s_{n-1}(\mathsf{H}) \xrightarrow{\mathsf{H}\,\mathsf{X}!\mathsf{m}} s(\mathsf{H}) \tag{A.4}$$

in $\mathsf{gw}(\mathsf{S}_1(\mathsf{H}), \mathsf{K})$. We have that ${}^{\mathsf{H}}_{\mathsf{K}}\!\lfloor s_{n-1} \in \mathcal{R}(\llbracket \mathsf{S}_1 \rrbracket)$ (by the inductive hypothesis). Also, since ${}^{\mathsf{H}}_{\mathsf{K}}\!\lfloor s_{n-1}(\mathsf{X}) = s_{n-1}(\mathsf{X})$, necessarily $s_{n-1}(\mathsf{X}) \xrightarrow{\mathsf{H}\,\mathsf{X}?\mathsf{m}} s(\mathsf{X}) = {}^{\mathsf{H}}_{\mathsf{K}}\!\lfloor s_{n-1}(\mathsf{X}) \in \mathsf{S}_1$ where the equalities hold by Definition 5.1. Hence, ${}^{\mathsf{H}}_{\mathsf{K}}\!\lfloor s_{n-1} \xrightarrow{\mathsf{H}\to\mathsf{X}\colon \mathsf{m}} {}^{\mathsf{H}}_{\mathsf{K}}\!\lfloor s_{n-1} = {}^{\mathsf{H}}_{\mathsf{K}}\!\lfloor s$ (by Definition 3.3) shows that ${}^{\mathsf{H}}_{\mathsf{K}}\!\lfloor s \in \mathcal{R}(\llbracket \mathsf{S}_1 \rrbracket)$.

We now show the compatibility part. The last transition in the run (A.1) must be preceded by a transition where the gateway machines $\mathsf{gw}(\mathsf{S}_1(\mathsf{H}), \mathsf{K})$ and $\mathsf{gw}(\mathsf{S}_2(\mathsf{K}), \mathsf{H})$ have exchanged message m; let i be the index of the last such transition in the run (A.4). Then, by (A.4) and !?-determinism, we also have

$$s_i(\mathsf{H}) = p \xrightarrow{\mathsf{K}\,\mathsf{H}?\mathsf{m}} s_{i+1}(\mathsf{H}) = q \in \mathsf{gw}(\mathsf{S}_1(\mathsf{H}), \mathsf{K}) \qquad \text{and} \qquad s_i(\mathsf{K}) \xrightarrow{\mathsf{K}\,\mathsf{H}!\mathsf{m}} s_{i+1}(\mathsf{K}) \in \mathsf{gw}(\mathsf{S}_2(\mathsf{K}), \mathsf{H})$$

with $\Gamma_{\mathsf{H},\mathsf{K}}(\mathsf{S}_1(\mathsf{H}), s_{i+1}(\mathsf{H})) \asymp \Gamma_{\mathsf{K},\mathsf{H}}(\mathsf{S}_2(\mathsf{K}), s_{i+1}(\mathsf{K}))$ by the inductive hypothesis, definition of compatibility, and !?-determinism. We now observe that, by definition of Γ,

$$\Gamma_{\mathsf{H},\mathsf{K}}(\mathsf{S}_1(\mathsf{H}), s_{n-1}(\mathsf{H})) = s(\mathsf{H}) = \Gamma_{\mathsf{H},\mathsf{K}}(\mathsf{S}_1(\mathsf{H}), s(\mathsf{H}))$$

Moreover, $s(\mathsf{K}) = s_{n-1}(\mathsf{K})$ because K is not involved in λ_n; also, 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 (A.1). In both cases, by definition of Γ, we have that $\Gamma_{\mathsf{K},\mathsf{H}}(\mathsf{S}_2(\mathsf{K}), s(\mathsf{K})) = \Gamma_{\mathsf{K},\mathsf{H}}(\mathsf{S}_2(\mathsf{K}), s_{n-1}(\mathsf{K})) = s_{i+1}(\mathsf{K})$. Then by the induction hypothesis

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

and the equalities above imply

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

$\boxed{\lambda_n = \mathsf{X}\to\mathsf{H}\colon \mathsf{m}}$ The case $\mathsf{X} = \mathsf{K}$, that is $\lambda_n = \mathsf{K}\to\mathsf{H}\colon \mathsf{m}$, is symmetric to the previous case and therefore omitted. So, assume that X is a participant of S_1; note that by construction $\mathsf{X} \neq \mathsf{H}$ (cf. Definition 3.3). Then in $\mathsf{gw}(\mathsf{S}_1(\mathsf{H}), \mathsf{K})$ there exists an exporting sequence $s_{n-1}(\mathsf{H}) \xrightarrow{\mathsf{X}\,\mathsf{H}?\mathsf{m}} s(\mathsf{H}) \xrightarrow{\tau} q \xrightarrow{\mathsf{H}\,\mathsf{K}!\mathsf{m}} r$ corresponding to a transition $s_{n-1}(\mathsf{H}) \xrightarrow{\mathsf{X}\,\mathsf{H}?\mathsf{m}} r \in \mathsf{S}_1$. We have that ${}^{\mathsf{H}}_{\mathsf{K}}\!\lfloor s_{n-1} \in \mathcal{R}(\llbracket \mathsf{S}_1 \rrbracket)$ by inductive hypothesis, and so is ${}^{\mathsf{H}}_{\mathsf{K}}\!\lfloor s_{n-1} \xrightarrow{\mathsf{X}\to\mathsf{H}\colon \mathsf{m}} {}^{\mathsf{H}}_{\mathsf{K}}\!\lfloor s$ since, by definition of left-projection, ${}^{\mathsf{H}}_{\mathsf{K}}\!\lfloor s(\mathsf{H}) = r$. The reachability of the right-projection of s immediately follows by the inductive hypothesis, since $s\lfloor^{\mathsf{K}}_{\mathsf{H}} = s_{n-1}\lfloor^{\mathsf{K}}_{\mathsf{H}}$.

We now show the compatibility condition. By definition, we have that

$$\Gamma_{\mathsf{H},\mathsf{K}}(\mathsf{S}_1(\mathsf{H}), s(\mathsf{H})) = \Gamma_{\mathsf{H},\mathsf{K}}(\mathsf{S}_1(\mathsf{H}), s_{n-1}(\mathsf{H}))$$

Moreover, we necessarily have that $s(\mathsf{K}) = s_{n-1}(\mathsf{K})$. Hence by the induction hypothesis,

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

The cases $\lambda_n = \mathsf{X}\to\mathsf{K}\colon \mathsf{m}$ and $\lambda_n = \mathsf{K}\to\mathsf{X}\colon \mathsf{m}$ are similar to the last two cases above. We conclude by noticing that the condition of absence of mixed states is actually not needed for the present proof to work. □

Appendix B. Proofs for Section 6 (What does Composability Buy Us?)

Proposition 6.1 (Weak lifting of τ!-systems)**.** *Let S_1 and S_2 be two (H,K)-composable and deadlock-free τ!-systems and let $\mathsf{S} = \mathsf{S}_1 {}^{\mathsf{H}\leftrightarrow\mathsf{K}} \mathsf{S}_2$. For all $s \in \mathcal{R}(\llbracket \mathsf{S} \rrbracket)$ if ${}^{\mathsf{H}}_{\mathsf{K}}\!\lfloor s \to$ or $s\lfloor^{\mathsf{K}}_{\mathsf{H}} \to$ then $s \to$.*

Proof. Without loss of generality, we show that if $\overset{\mathsf{H}}{\mathsf{K}}\rfloor s \xrightarrow{\lambda}$ for a certain λ then $s \xrightarrow{\lambda}$. We proceed by case analysis on λ noticing that any participant involved in λ belongs to S_1. Recall that, by Proposition 5.9, $\overset{\mathsf{H}}{\mathsf{K}}\rfloor s \in \mathcal{R}(\llbracket \mathsf{S}_1 \rrbracket)$.

$\boxed{\lambda = \mathsf{Y}{\to}\mathsf{X}\colon \mathsf{m} \text{ with } \mathsf{H} \notin \{\mathsf{X}, \mathsf{Y}\}}$ Then, by Definition 5.1, we immediately get $s \xrightarrow{\lambda}$.

$\boxed{\lambda = \mathsf{H}{\to}\mathsf{X}\colon \mathsf{m}}$ By the definition of the projection operation (cf. Definition 5.1), $\overset{\mathsf{H}}{\mathsf{K}}\rfloor s(\mathsf{H})$ has an outgoing output transition with label $\mathsf{H}\,\mathsf{X}!\mathsf{m}$ and $\overset{\mathsf{H}}{\mathsf{K}}\rfloor s(\mathsf{X}) = s(\mathsf{X})$ has a corresponding input transition by hypothesis. Hence, $s \xrightarrow{\lambda}$ by Definition 3.3.

$\boxed{\lambda = \mathsf{X}{\to}\mathsf{H}\colon \mathsf{m}}$ This means, by definition of synchronisation (Definition 3.3), that $\overset{\mathsf{H}}{\mathsf{K}}\rfloor s(\mathsf{X}) \xrightarrow{\mathsf{X}\,\mathsf{H}!\mathsf{m}}$ and $\overset{\mathsf{H}}{\mathsf{K}}\rfloor s(\mathsf{H}) \xrightarrow{\mathsf{X}\,\mathsf{H}?\mathsf{m}}$. Since necessarily $\mathsf{X} \neq \mathsf{K}$, we have $\overset{\mathsf{H}}{\mathsf{K}}\rfloor s(\mathsf{X}) = s(\mathsf{X})$. Now, from $\overset{\mathsf{H}}{\mathsf{K}}\rfloor s(\mathsf{H}) \xrightarrow{\mathsf{X}\,\mathsf{H}?\mathsf{m}}$ and Definition 5.1, we can infer that one of the following cases are possible in $\mathsf{S}_1 {}^{\mathsf{H}\not\leftrightarrow\mathsf{K}} \mathsf{S}_2$ for what concerns the state $s(\mathsf{H})$:

(i) $\overset{\mathsf{H}}{\mathsf{K}}\rfloor s(\mathsf{H}) = s(\mathsf{H}) \xrightarrow{\mathsf{X}\,\mathsf{H}?\mathsf{m}}$

(ii) $s(\mathsf{H}) \xrightarrow{\tau} \xrightarrow{\mathsf{H}\,\mathsf{K}!\mathsf{n}} \overset{\mathsf{H}}{\mathsf{K}}\rfloor s(\mathsf{H}) \xrightarrow{\mathsf{X}\,\mathsf{H}?\mathsf{m}}$ (since $s(\mathsf{H})$ is the "pre-τ" internal state of an exporting segment)

(iii) $s(\mathsf{H}) \xrightarrow{\mathsf{H}\,\mathsf{K}!\mathsf{n}} \overset{\mathsf{H}}{\mathsf{K}}\rfloor s(\mathsf{H}) \xrightarrow{\mathsf{X}\,\mathsf{H}?\mathsf{m}}$ (since $s(\mathsf{H})$ is the "post-τ" internal state of an importing segment)

Case (i) We get easily the thesis. In fact, from $s(\mathsf{X}) = \overset{\mathsf{H}}{\mathsf{K}}\rfloor s(\mathsf{X}) \xrightarrow{\mathsf{X}\,\mathsf{H}!\mathsf{m}}$ we get $s \xrightarrow{\mathsf{X}\to\mathsf{H}\colon \mathsf{m}}$.
Case (ii) This is trivial, since, by definition of synchronous semantics (cf. Definition 3.3), we have that $s \xrightarrow{\tau}$.
Case (iii) In this case we have in particular that

$$p \xrightarrow{\mathsf{Y}\,\mathsf{H}?\mathsf{n}} p' \xrightarrow{\tau} s(\mathsf{H}) \xrightarrow{\mathsf{H}\,\mathsf{K}!\mathsf{n}} \overset{\mathsf{H}}{\mathsf{K}}\rfloor s(\mathsf{H}) \xrightarrow{\mathsf{X}\,\mathsf{H}?\mathsf{m}}$$

where $p = \Gamma_{\mathsf{H},\mathsf{K}}(\mathsf{S}_1(\mathsf{H}), s(\mathsf{H}))$. Now, by Proposition 5.9, we obtain $p \asymp \Gamma_{\mathsf{K},\mathsf{H}}(\mathsf{S}_2(\mathsf{K}), s(\mathsf{K}))$. Since in $\mathsf{S}_1(\mathsf{H})$ we have

$$p \xrightarrow{\mathsf{Y}\,\mathsf{H}?\mathsf{n}} \overset{\mathsf{H}}{\mathsf{K}}\rfloor s(\mathsf{H})$$

we can use the definition of compatibility to get that either

$$\Gamma_{\mathsf{K},\mathsf{H}}(\mathsf{S}_2(\mathsf{K}), s(\mathsf{K})) \xrightarrow{\mathsf{K}\,\mathsf{B}!\mathsf{n}} \quad \text{or} \quad \Gamma_{\mathsf{K},\mathsf{H}}(\mathsf{S}_2(\mathsf{K}), s(\mathsf{K})) \xrightarrow{\tau} q \xrightarrow{\mathsf{K}\,\mathsf{B}!\mathsf{n}}$$

with $p \asymp q$ for certain B and q. The former case cannot ever occur by the definition of Γ. So, let us consider the latter possibility. By definition of Γ one of the following cases is possible

$$\Gamma_{\mathsf{K},\mathsf{H}}(\mathsf{S}_2(\mathsf{K}), s(\mathsf{K})) = s(\mathsf{K}) \xrightarrow{\mathsf{H}\,\mathsf{K}?\mathsf{n}} p' \xrightarrow{\tau} q \xrightarrow{\mathsf{K}\,\mathsf{B}!\mathsf{n}} \tag{B.1}$$

$$s(\mathsf{K}) \xrightarrow{\tau} \xrightarrow{\mathsf{K}\,\mathsf{C}!\mathsf{z}} \Gamma_{\mathsf{K},\mathsf{H}}(\mathsf{S}_2(\mathsf{K}), s(\mathsf{K})) \tag{B.2}$$

$$s(\mathsf{K}) \xrightarrow{\mathsf{K}\,\mathsf{C}!\mathsf{z}} \Gamma_{\mathsf{K},\mathsf{H}}(\mathsf{S}_2(\mathsf{K}), s(\mathsf{K})) \tag{B.3}$$

Case (B.1) This case is easy: we had $s(\mathsf{H}) \xrightarrow{\mathsf{H}\,\mathsf{K}!\mathsf{n}}$ and then $s \xrightarrow{\mathsf{H}\to\mathsf{K}\colon \mathsf{n}}$ by definition of synchronisation (Definition 3.3).
Case (B.2) The thesis descends easily since $s \xrightarrow{\tau}$ by definition of synchronisation.
Case (B.3) We observe that, by definition of projections, we have in particular that in $(\mathsf{S}_1 {}^{\mathsf{H}\not\leftrightarrow\mathsf{K}} \mathsf{S}_2)(\mathsf{K})$

$$s\lfloor^{\mathsf{K}}_{\mathsf{H}}(\mathsf{K}) \xrightarrow{\mathsf{H}\,\mathsf{K}?\mathsf{z}} p_2' \xrightarrow{\tau} s(\mathsf{K}) \xrightarrow{\mathsf{K}\,\mathsf{C}!\mathsf{z}}$$

for a certain p_2'. Moreover, we observe that, again by definition of projection, in $\mathsf{S}_2(\mathsf{K})$ we have

$$s\lfloor^{\mathsf{K}}_{\mathsf{H}}(\mathsf{K}) \xrightarrow{\tau} s(\mathsf{K}) \xrightarrow{\mathsf{K}\,\mathsf{C}!\mathsf{z}}$$

We now have the following cases to consider, noticing that, by the absence of mixed states, a transition of the form $s\lfloor^{\mathsf{K}}_{\mathsf{H}}(\mathsf{K}) \xrightarrow{\_\mathsf{K}?\_}$ cannot occur in $\mathsf{S}_2(\mathsf{K})$.

$s(\mathsf{D})\lfloor^{\mathsf{K}}_{\mathsf{H}} \xrightarrow{\tau}$ in S_2 for some participant $\mathsf{D} \neq \mathsf{K}$. This means that, by definition of projection, $s(\mathsf{D})\lfloor^{\mathsf{K}}_{\mathsf{H}} = s(\mathsf{D})$. So, by definition of synchronisation we get the thesis, namely $s \xrightarrow{\tau}$.

$s\lfloor^{\mathsf{K}}_{\mathsf{H}}(\mathsf{C}) \xrightarrow{\mathsf{K}\,\mathsf{C}?\mathsf{z}}$ in S_2. We easily get the thesis since, being $\mathsf{C} \neq \mathsf{K}$, we have $s\lfloor^{\mathsf{K}}_{\mathsf{H}}(\mathsf{C}) = s(\mathsf{K})$ and hence $s \xrightarrow{\mathsf{K}\to\mathsf{C}:\,\mathsf{z}}$.

Otherwise, let $\hat{s}$ be the configuration of S_2 such that $\hat{s}(\mathsf{K}) = s(\mathsf{K})$ and $\hat{s}(\mathsf{X}) = s\lfloor^{\mathsf{K}}_{\mathsf{H}}(\mathsf{X})$ for any $\mathsf{X} \neq \mathsf{K}$. So, by definition of synchronisation, $s\lfloor^{\mathsf{K}}_{\mathsf{H}} \xrightarrow{\tau} \hat{s}$ and hence $\hat{s} \in \mathcal{R}(\llbracket \mathsf{S}_2 \rrbracket)$. We observe that $\hat{s}(\mathsf{K}) \xrightarrow{\mathsf{K}\,\mathsf{C}!\mathsf{z}}$ is the unique transition out of $\hat{s}(\mathsf{K})$ by definition of τ!-CFSMs.[6] Now, we can recur to deadlock-freedom[7] of S_2 in order to infer that necessarily either

$$\hat{s} \xrightarrow{\tau} \quad \text{or} \quad \hat{s} \xrightarrow{\mathsf{E}\to\mathsf{F}:\,\mathsf{y}}$$

for some $\mathsf{E}, \mathsf{F} \neq \mathsf{K}$, otherwise $\hat{s}$ would be a deadlock for S_2. In the first case, we have necessarily that $\hat{s}(\mathsf{D}) \xrightarrow{\tau}$ for some $\mathsf{D} \neq \mathsf{K}$. So, since $\hat{s}(\mathsf{D}) = s\lfloor^{\mathsf{K}}_{\mathsf{H}}(\mathsf{D}) = s(\mathsf{D})$, we get $s \xrightarrow{\tau}$ and we are done. In the second case, we could have

- $\mathsf{E}, \mathsf{F} \neq \mathsf{K}$. Hence, from $\hat{s}(\mathsf{X}) = s\lfloor^{\mathsf{K}}_{\mathsf{H}}(\mathsf{X}) = s(\mathsf{X})$ for any $\mathsf{X} \neq \mathsf{K}$, we get that $s \xrightarrow{\mathsf{E}\to\mathsf{F}:\,\mathsf{y}}$ and we are done.
- $\mathsf{E} = \mathsf{K}, \mathsf{F} = \mathsf{C}$ and $\mathsf{y} = \mathsf{z}$. In this case, since $\hat{s}(\mathsf{K}) = s(\mathsf{K})$ and $\hat{s}(\mathsf{X}) = s\lfloor^{\mathsf{K}}_{\mathsf{H}}(\mathsf{X})$ for any $\mathsf{X} \neq \mathsf{K}$, we get $s \xrightarrow{\mathsf{K}\to\mathsf{C}:\,\mathsf{z}}$ and we are done.

We conclude by observing that the two above are the only possibilities for E, F and y, since, as previously mentioned, there is only one outgoing transition from $\hat{s}(\mathsf{K})$, namely $\hat{s}(\mathsf{K}) \xrightarrow{\mathsf{K}\,\mathsf{C}!\mathsf{z}}$.

$\boxed{\lambda = \tau}$ Let X be the participant of S_1 involved in such transition.

If $\mathsf{X} \neq \mathsf{H}$ then $s(\mathsf{X}) =^{\mathsf{H}}_{\mathsf{K}} s(\mathsf{X})$ by definition of projection (cf. Definition 5.1); hence $s \xrightarrow{\tau}$ by definition of synchronisation (cf. Definition 3.3).

If $\mathsf{X} = \mathsf{H}$ then we proceed by case analysis on the state $s(\mathsf{H})$ of the gateway $\mathsf{gw}(\mathsf{S}_1(\mathsf{H}), \mathsf{K})$ in s. If $s(\mathsf{H})$ is a fresh state of an exporting or importing segment in the gateway $\mathsf{gw}(\mathsf{S}_1(\mathsf{H}), \mathsf{K})$ with an enabled τ-transition then $s \xrightarrow{\tau}$. Otherwise, one of the following cases holds:

(a) $q \xrightarrow{\mathsf{K}\,\mathsf{H}?\mathsf{m}} q'' \xrightarrow{\tau} q' \xrightarrow{\mathsf{H}\,\mathsf{A}!\mathsf{m}} r$ in $\mathsf{gw}(\mathsf{S}_1(\mathsf{H}), \mathsf{K})$ for some fresh state q'' and $^{\mathsf{H}}_{\mathsf{K}}\rfloor s(\mathsf{H}) \in \{q, q', r\}$

(b) $q \xrightarrow{\mathsf{A}\,\mathsf{H}?\mathsf{m}} q' \xrightarrow{\tau} q'' \xrightarrow{\mathsf{H}\,\mathsf{K}!\mathsf{m}} r$ in $\mathsf{gw}(\mathsf{S}_1(\mathsf{H}), \mathsf{K})$ for some fresh states q' and q'' and $^{\mathsf{H}}_{\mathsf{K}}\rfloor s(\mathsf{H}) \in \{q, r\}$

where the segment in (a) is induced by the transitions $q \xrightarrow{\tau} q' \xrightarrow{\mathsf{H}\,\mathsf{A}!\mathsf{m}} r$ in $\mathsf{S}_1(\mathsf{H})$ and the segment in (b) is induced by the transition $q \xrightarrow{\mathsf{A}\,\mathsf{H}?\mathsf{m}} r$ in $\mathsf{S}_1(\mathsf{H})$. Necessarily, we are either in case (a) and $^{\mathsf{H}}_{\mathsf{K}}\rfloor s(\mathsf{H}) = q$ or in case (b) and $^{\mathsf{H}}_{\mathsf{K}}\rfloor s(\mathsf{H}) = r$ otherwise $^{\mathsf{H}}_{\mathsf{K}}\rfloor s(\mathsf{H}) \not\xrightarrow{\tau}$ contrary to the hypothesis that H is involved in $^{\mathsf{H}}_{\mathsf{K}}\rfloor s \xrightarrow{\tau}$. In the former case the thesis follows because $\Gamma_{\mathsf{H},\mathsf{K}}(\mathsf{S}_1(\mathsf{H}), s(\mathsf{H})) \asymp \Gamma_{\mathsf{K},\mathsf{H}}(\mathsf{S}_2(\mathsf{K}), s(\mathsf{K}))$ (by Proposition 5.9) and therefore $s \xrightarrow{\mathsf{K}\to\mathsf{H}:\,\mathsf{m}}$. In the other case either $s(\mathsf{H}) = q''$ or $s(\mathsf{H}) = r$ by definition of projection. If $s(\mathsf{H}) = q''$ then $s \xrightarrow{\mathsf{H}\to\mathsf{K}:\,\mathsf{m}}$ since $\Gamma_{\mathsf{H},\mathsf{K}}(\mathsf{S}_1(\mathsf{H}), s(\mathsf{H})) \asymp \Gamma_{\mathsf{K},\mathsf{H}}(\mathsf{S}_2(\mathsf{K}), s(\mathsf{K}))$ (by Proposition 5.9). If $s(\mathsf{H}) = r$ then the proof proceeds as in the case (a).

□

Recall that (H, K)-composability requires absence of mixed states and ?!-determinism.

[6] This is essentially why we do not need the sequentiality condition that will instead be needed in Proposition 8.5, the CFSMs version of present proposition.

[7] Notice that in the present proposition we are only expecting to find a transition out of s. If we had needed, instead, to use this proposition to "construct" a run of S_1 out of a run of $\mathsf{S}_1 \overset{\mathsf{H}\leftrightarrow\mathsf{K}}{\cdot} \mathsf{S}_2$, we should have been able to get a transition $\mathsf{H}\to\mathsf{A}:\,\mathsf{m}$ after the τ transition, whereas here we can only go on with a transition $\mathsf{H}\to\mathsf{B}:\,\mathsf{b}$. Only sequentiality could guarantee that. In fact such a requirement is needed in Proposition 8.3.

Proposition B.1 (Enabledness in composed system implies enabledness in projections)**.** *Let* $\mathsf{S} = \mathsf{S}_1 {}^{\mathsf{H}\leftrightarrow\mathsf{K}} \mathsf{S}_2$ *be the composition of* S_1 *and* S_2*, two* (H,K)*-composable* τ!*-systems. For all* $s \in \mathcal{R}(\llbracket \mathsf{S} \rrbracket)$*, if* s *enables a participant* $\mathsf{A} \in \mathsf{S}$*, then* ${}^{\mathsf{H}}_{\mathsf{K}}\rfloor s$ *enables a participant in* S_1 *or* $s\lfloor^{\mathsf{K}}_{\mathsf{H}}$ *enables a participant in* S_2*.*

Proof. Without loss of generality, we can assume $\mathsf{A} \in \mathsf{S}_1$, the case $\mathsf{A} \in \mathsf{S}_2$ being similar. Note that both ${}^{\mathsf{H}}_{\mathsf{K}}\rfloor s$ and $s\lfloor^{\mathsf{K}}_{\mathsf{H}}$ are reachable in, respectively, S_1 and S_2 by Proposition 5.9.

If $\mathsf{A} \neq \mathsf{H}$ then any transition of A enabled in s is also enabled in ${}^{\mathsf{H}}_{\mathsf{K}}\rfloor s$ since ${}^{\mathsf{H}}_{\mathsf{K}}\rfloor s(\mathsf{A}) = s(\mathsf{A})$ by Definition 5.1. If $\mathsf{A} = \mathsf{H}$, then either of the following three cases occurs:

H has enabled an input. Assume that the input is from K.

Then, by construction (Definition 5.1), ${}^{\mathsf{H}}_{\mathsf{K}}\rfloor s(\mathsf{H})$ has a τ-transition enabled in $\mathsf{S}_1(\mathsf{H})$.

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

H has enabled an output. If the receiver of the output from H is a participant of S_1 then s is the internal state of an importing sequence in $s(\mathsf{H})$ by definition of gateway. Therefore, by configuration projection, ${}^{\mathsf{H}}_{\mathsf{K}}\rfloor s(\mathsf{H})$ either has an outgoing output or a τ-transition.

If the receiver of the output from H is K necessarily $s(\mathsf{H})$ has an importing sequence

$$p \xrightarrow{\mathsf{B}\,\mathsf{H}?\mathsf{m}} p' \xrightarrow{\tau} s(\mathsf{H}) \xrightarrow{\mathsf{H}\,\mathsf{K}!\mathsf{m}} r$$

corresponding to a transition $p \xrightarrow{\mathsf{B}\,\mathsf{H}?\mathsf{m}} r$ where $r = {}^{\mathsf{H}}_{\mathsf{K}}\rfloor s(\mathsf{H})$. We can now appeal to Proposition 5.9 and definition of Γ in order to get

$$p = \Gamma_{\mathsf{H},\mathsf{K}}(\mathsf{S}_1(\mathsf{H}), s(\mathsf{H})) \asymp \Gamma_{\mathsf{K},\mathsf{H}}(\mathsf{S}_2(\mathsf{K}), s(\mathsf{K}))$$

By definition of compatibility (Definition 4.6), from $p \xrightarrow{\mathsf{B}\,\mathsf{H}?\mathsf{m}} r$ there is a transition in S_2 of the form

$$\Gamma_{\mathsf{K},\mathsf{H}}(\mathsf{S}_2(\mathsf{K}), s(\mathsf{K})) \xrightarrow{\mathsf{K}\,\mathsf{C}!\mathsf{m}} q$$

We hence get what we needed by observing that $\Gamma_{\mathsf{K},\mathsf{H}}(\mathsf{S}_2(\mathsf{K}), s(\mathsf{K})) = s(\mathsf{K})$, in fact, K is enabled in $s\lfloor^{\mathsf{K}}_{\mathsf{H}}$ since it has an outgoing τ-transition.

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 S with $\mathsf{X} = \mathsf{K}$ or $\mathsf{X} \neq \mathsf{H}$ participant of S_1. In the former case we can reason as in the previous case when H outputs to K. Otherwise, ${}^{\mathsf{H}}_{\mathsf{K}}\rfloor s(\mathsf{H})$ has an enabled τ-transition in $\mathsf{S}_1(\mathsf{H})$ by Definition 5.1. □

Appendix C. Proofs for Section 8 (Raise the Bet with Sequentiality and Win the Pot)

Proposition 8.2 (τ delaying)**.** *Given a* τ!*-system* S*, for all* $s \in \mathcal{R}(\llbracket \mathsf{S} \rrbracket)$*, if* $s \xrightarrow{\lambda} s'$ *and* $\lambda \neq \tau$ *then there is* $\hat{s} \in \mathcal{R}(\llbracket \mathsf{S} \rrbracket)$ *such that* $\hat{s} \xrightarrow{\tau} s \xrightarrow{\lambda} s'$ *and the sender of* λ *is involved in* $\hat{s} \xrightarrow{\tau} s$*.*

Proof. Let $\lambda = \mathsf{A}{\to}\mathsf{B}\colon \mathsf{m}$ and

$$s_0 \xrightarrow{\lambda_1} s_1 \cdots s_{n-1} \xrightarrow{\lambda_n} s_n = s \xrightarrow{\lambda} s' \tag{C.1}$$

be a run from the initial configuration of S to s' passing through s. By the semantics of τ!-systems (cf. Definition 3.3) there is an index $0 \leq h < n$ such that $s_h \xrightarrow{\tau} s_{h+1}$ is a τ-transition involving A; let i be the largest of such indexes. Necessarily, $s_i(\mathsf{A}) \xrightarrow{\tau} s_{i+1}(\mathsf{A}) \xrightarrow{\mathsf{A}\,\mathsf{B}!\mathsf{m}} s'(\mathsf{A}) \in \mathsf{S}(\mathsf{A})$ and, for all $i+1 \leq h < n$, $s_h(\mathsf{A}) = s_{i+1}(\mathsf{A})$ since A cannot be involved in transitions other than the last one in (C.1). Therefore, the following run exists in $\llbracket \mathsf{S} \rrbracket$:

$$s_i \xrightarrow{\lambda_{i+1}} s'_{i+1} \cdots s'_{n-1} \xrightarrow{\lambda_n} s'_n$$

where $s'_h = s_h[\mathsf{A} \mapsto s_i(\mathsf{A})]$ for all $i+1 \leq h \leq n$. Hence, the thesis follows by observing that $s'_n \xrightarrow{\tau} s$ with A involved in the τ-transition. □

Lemma C.1 (Interaction lifting)**.** *Let* S_1 *and* S_2 *be two* (H,K)*-composable and lock-free communicating or* $\tau!$*-systems with* H *and* K *sequential and let* $\mathsf{S} = \mathsf{S}_1 {}^{\mathsf{H}}\!\!\not\leftrightarrow^{\mathsf{K}} \mathsf{S}_2$*. For all* $s \in \mathcal{R}([\![\mathsf{S}]\!])$ *and interaction labels* $\lambda \neq \tau$*, if* ${}^{\mathsf{H}}_{\mathsf{K}}\rfloor s \xrightarrow{\lambda} \bar{s}$ *in* $[\![\mathsf{S}_1]\!]$ *then there is a configuration* $\hat{s} \in \mathcal{R}([\![\mathsf{S}]\!], s)$ *such that* $\hat{s} \xrightarrow{\lambda} s'$ *and* ${}^{\mathsf{H}}_{\mathsf{K}}\rfloor s' = \bar{s}$*.*

The same holds for the right-projection.

Proof. Let $\lambda = \mathsf{X}{\to}\mathsf{Y}\colon \mathsf{m}$ with $\mathsf{X},\mathsf{Y} \in \mathsf{ptp}(\mathsf{S}_1)$. For some states q and q',

$$ {}^{\mathsf{H}}_{\mathsf{K}}\rfloor s(\mathsf{X}) \xrightarrow{\mathsf{X}\,\mathsf{Y}!\mathsf{m}} q \in \mathsf{S}_1(\mathsf{X}) \qquad \text{and} \qquad {}^{\mathsf{H}}_{\mathsf{K}}\rfloor s(\mathsf{Y}) \xrightarrow{\mathsf{X}\,\mathsf{Y}?\mathsf{m}} q' \in \mathsf{S}_1(\mathsf{Y}) $$

and necessarily $\bar{s} = ({}^{\mathsf{H}}_{\mathsf{K}}\rfloor s)[\mathsf{X} \mapsto q, \mathsf{Y} \mapsto q']$ by the definition of synchronous semantics (cf. Definition 3.3). We have the following three cases depending on whether H is not or is involved in λ.

$\boxed{\mathsf{H} \notin \mathsf{ptp}(\lambda)}$ The left-projection operation leaves unchanged the states of non-interface machines (cf. Definition 5.1), hence $s(\mathsf{X}) = {}^{\mathsf{H}}_{\mathsf{K}}\rfloor s(\mathsf{X})$ and $s(\mathsf{Y}) = {}^{\mathsf{H}}_{\mathsf{K}}\rfloor s(\mathsf{Y})$. Hence $s \xrightarrow{\lambda} s'$ and we have the thesis.

$\boxed{\mathsf{H} = \mathsf{X}}$ If the state $s(\mathsf{H})$ is the internal non-fresh state of an importing segment of the gateway $\mathsf{gw}(\mathsf{S}_1(\mathsf{H}), \mathsf{K})$. Hence we have $s(\mathsf{H}) = {}^{\mathsf{H}}_{\mathsf{K}}\rfloor s(\mathsf{H})$ and, as before, $s(\mathsf{Y}) = {}^{\mathsf{H}}_{\mathsf{K}}\rfloor s(\mathsf{Y})$ since the left-projection operation leaves unchanged non-fresh states. The thesis follows by observing that $s \xrightarrow{\lambda} s'$ with $s' = s[\mathsf{X} \mapsto q, \mathsf{Y} \mapsto q']$ and, since q and q' are non-fresh states, we have ${}^{\mathsf{H}}_{\mathsf{K}}\rfloor s' = ({}^{\mathsf{H}}_{\mathsf{K}}\rfloor s)[\mathsf{X} \mapsto q, \mathsf{Y} \mapsto q'] = \bar{s}$.

For communicating systems we have to consider also the case that $s(\mathsf{H})$ is the starting state of an importing segment; that is $s(\mathsf{H}) \xrightarrow{\mathsf{K}\,\mathsf{H}?\mathsf{m}} q' \in \mathsf{S}_1(\mathsf{H})$. In this case, by Proposition 5.9 we have

$$ s(\mathsf{H}) = \Gamma_{\mathsf{H},\mathsf{K}}(\mathsf{S}_1(\mathsf{H}), s(\mathsf{H})) \quad \asymp \quad \Gamma_{\mathsf{K},\mathsf{H}}(\mathsf{S}_2(\mathsf{K}), s(\mathsf{K})) $$

and, by lock-freedom of S_2, s can reach a transition s'' such that

$$ s'' \xrightarrow{\mathsf{K}\to\mathsf{H}\colon \mathsf{m}} s''', \qquad {}^{\mathsf{H}}_{\mathsf{K}}\rfloor s'' = {}^{\mathsf{H}}_{\mathsf{K}}\rfloor s, \qquad \text{and} \qquad {}^{\mathsf{H}}_{\mathsf{K}}\rfloor s''' = ({}^{\mathsf{H}}_{\mathsf{K}}\rfloor s)[\mathsf{H} \mapsto q'] $$

Hence, $s''' \xrightarrow{\lambda} s'$ with ${}^{\mathsf{H}}_{\mathsf{K}}\rfloor s' = \bar{s}$.

$\boxed{\mathsf{H} = \mathsf{Y}}$ If gateway $s(\mathsf{H})$ is in the starting state of an exporting segment then

$$ s(\mathsf{H}) = {}^{\mathsf{H}}_{\mathsf{K}}\rfloor s(\mathsf{H}) \qquad \text{and} \qquad s(\mathsf{X}) = {}^{\mathsf{H}}_{\mathsf{K}}\rfloor s(\mathsf{X}) $$

which imply $s \xrightarrow{\lambda} s'$ with $s'(\mathsf{H})$ the first fresh state of the exporting transition corresponding to ${}^{\mathsf{H}}_{\mathsf{K}}\rfloor s(\mathsf{H}) \xrightarrow{\mathsf{X}\,\mathsf{H}?\mathsf{m}} q \in \mathsf{S}_1(\mathsf{H})$. The thesis then follows since ${}^{\mathsf{H}}_{\mathsf{K}}\rfloor s' = ({}^{\mathsf{H}}_{\mathsf{K}}\rfloor s)[\mathsf{H} \mapsto q', \mathsf{X} \mapsto q]$ by the definition of left-projection.

Suppose now that $s(\mathsf{H})$ is an internal state of an exporting segment. Then

(a) either $s(\mathsf{H}) \xrightarrow{\tau} q'' \xrightarrow{\mathsf{H}\,\mathsf{K}!\mathsf{m}} r$ (for $\tau!$-systems only)

(b) or $s(\mathsf{H}) \xrightarrow{\mathsf{H}\,\mathsf{K}!\mathsf{m}} r$ (for communicating and $\tau!$-systems)

In case (a), $s \xrightarrow{\tau} s[\mathsf{H} \mapsto q'']$ and by Proposition 5.9

$$ (s[\mathsf{H} \mapsto q''])(\mathsf{K}) = \Gamma_{\mathsf{K},\mathsf{H}}(\mathsf{S}_2(\mathsf{K}), s(\mathsf{K})) \asymp \Gamma_{\mathsf{H},\mathsf{K}}(\mathsf{S}_1(\mathsf{H}), q'') = {}^{\mathsf{H}}_{\mathsf{K}}\rfloor s(\mathsf{H}) $$

Hence, by lock freedom of S_2 and the sequentiality of $\mathsf{S}_2(\mathsf{K})$ we have

$$ s[\mathsf{H} \mapsto q''] \xrightarrow{\mathsf{K}\to\mathsf{H}\colon \mathsf{m}} \hat{s} \qquad \text{with} \qquad {}^{\mathsf{H}}_{\mathsf{K}}\rfloor \hat{s} = {}^{\mathsf{H}}_{\mathsf{K}}\rfloor s $$

by the definition of left-projection. This yields the thesis since ${}^{\mathsf{H}}_{\mathsf{K}}\rfloor s(\mathsf{H}) \xrightarrow{\mathsf{X}\,\mathsf{H}?\mathsf{m}} q \in \mathsf{S}_1(\mathsf{H})$.

The proof in case (b) is similar to case (a) but for the fact that s does not do a τ-transition. □

Proposition 8.3 (Transitions lifting)**.** *Let* S_1 *and* S_2 *be two* (H,K)*-composable and lock-free communicating or* $\tau!$*-systems with* H *and* K *sequential and let* $\mathsf{S} = \mathsf{S}_1 {}^{\mathsf{H}}\!\!\not\leftrightarrow^{\mathsf{K}} \mathsf{S}_2$*. For all* $s \in \mathcal{R}([\![\mathsf{S}]\!])$ *and labels* λ*, if* ${}^{\mathsf{H}}_{\mathsf{K}}\rfloor s \xrightarrow{\lambda} \bar{s}$ *in* $[\![\mathsf{S}_1]\!]$ *then there is a configuration* $\hat{s} \in \mathcal{R}([\![\mathsf{S}]\!], s)$ *such that* $\hat{s} \xrightarrow{\lambda} s'$ *and* ${}^{\mathsf{H}}_{\mathsf{K}}\rfloor s' = \bar{s}$*.*

The same holds for the right-projection.

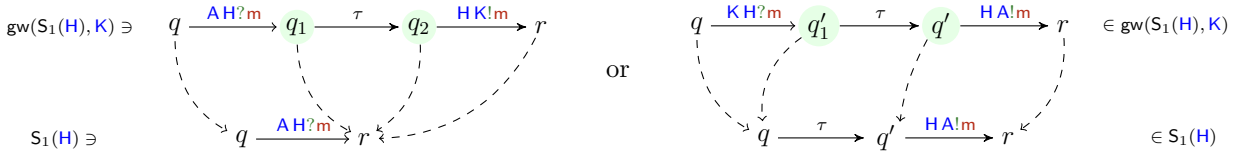
Proof. If $\lambda \neq \tau$ the thesis follows by Lemma C.1, hence we only have to consider τ-transitions of τ!-systems.

Let $\overset{\mathsf{H}}{\mathsf{K}}\rfloor s \xrightarrow{\tau} \bar{s}$ and X be the participant of S_1 involved in this τ-transition.

If $\mathsf{X} \neq \mathsf{H}$ then there is a transition $\overset{\mathsf{H}}{\mathsf{K}}\rfloor s(\mathsf{X}) = s(\mathsf{X}) \xrightarrow{\tau} q'$ in the machine $\mathsf{S}_1(\mathsf{X})$ and $\bar{s} = (\overset{\mathsf{H}}{\mathsf{K}}\rfloor s)[\mathsf{X} \mapsto q']$; hence, $s \xrightarrow{\tau} s[\mathsf{X} \mapsto q']$ and $\overset{\mathsf{H}}{\mathsf{K}}\rfloor(s[\mathsf{X} \mapsto q']) = \bar{s}$.

When $\mathsf{X} = \mathsf{H}$ the proof is more complex. We proceed by case analysis depending on whether $s(\mathsf{H})$ is or is not an internal state of an exporting or an importing segment in $\mathsf{gw}(\mathsf{S}_1(\mathsf{H}), \mathsf{K})$.

Case $s(\mathsf{H})$ internal By our gateway construction (cf. Definition 4.1), we have



where the highlighted states on the left (resp. right) are those internal on an exporting (resp. importing) segment. Let us consider each possibility.

$\boxed{s(\mathsf{H}) = q_1}$ By definition of τ!-CFSMs, there are a participant $\mathsf{B} \neq \mathsf{H}$ of S_1 and a message n such that

$$r \xrightarrow{\tau} r_1 \xrightarrow{\mathsf{H}\,\mathsf{B}!\mathsf{n}} r_2 \qquad \text{are transition in } \mathsf{S}_1(\mathsf{H}).$$

and we have

$$\overset{\mathsf{H}}{\mathsf{K}}\rfloor s(\mathsf{H}) = r, \qquad \text{and} \qquad \bar{s} = (\overset{\mathsf{H}}{\mathsf{K}}\rfloor s)[\mathsf{H} \mapsto r_1] \tag{C.2}$$

Again by our gateway construction,

$$r \xrightarrow{\mathsf{K}\,\mathsf{H}?\mathsf{n}} r_1' \xrightarrow{\tau} r_1 \xrightarrow{\mathsf{H}\,\mathsf{B}!\mathsf{n}} r_2 \qquad \text{are transitions in } \mathsf{gw}(\mathsf{S}_1(\mathsf{H}), \mathsf{K}). \tag{C.3}$$

By Proposition 5.9 and the definition of gateway embedding (cf. Definition 5.6)

$$\Gamma_{\mathsf{H},\mathsf{K}}(\mathsf{S}_1(\mathsf{H}), s(\mathsf{H})) = q \asymp \Gamma_{\mathsf{K},\mathsf{H}}(\mathsf{S}_2(\mathsf{K}), s(\mathsf{K}))$$

Therefore, by the sequentiality of K and the lock freedom of S_2 there is a configuration $s_1 \in \mathcal{R}([\![\mathsf{S}]\!], s)$ reachable from s such that

$$s_1 \xrightarrow{\tau} \xrightarrow{\mathsf{H} \longrightarrow \mathsf{K}:\,\mathsf{m}} s'' \qquad \text{where } \mathsf{H} \text{ is involved in the } \tau \text{ transition} \tag{C.4}$$

and $\overset{\mathsf{H}}{\mathsf{K}}\rfloor s'' = (\overset{\mathsf{H}}{\mathsf{K}}\rfloor s)[\mathsf{H} \mapsto r]$. Also, $\Gamma_{\mathsf{H},\mathsf{K}}(\mathsf{S}_1(\mathsf{H}), r) \asymp \Gamma_{\mathsf{K},\mathsf{H}}(\mathsf{S}_2(\mathsf{K}), s''(\mathsf{K}))$ (by Proposition 5.9, Equation (C.3) above and Proposition 8.2) imply

$$s'' \xrightarrow{\tau} \xrightarrow{\mathsf{K} \longrightarrow \mathsf{H}:\,\mathsf{n}} \hat{s} \xrightarrow{\tau} s' \qquad \text{with } \mathsf{K} \text{ (resp. } \mathsf{H}) \text{ is involved in the first (resp. second) } \tau \text{ transition}$$

where $\overset{\mathsf{H}}{\mathsf{K}}\rfloor s' = \overset{\mathsf{H}}{\mathsf{K}}\rfloor s''[\mathsf{H} \mapsto r_1] = (\overset{\mathsf{H}}{\mathsf{K}}\rfloor s)[\mathsf{H} \mapsto r_1]$.

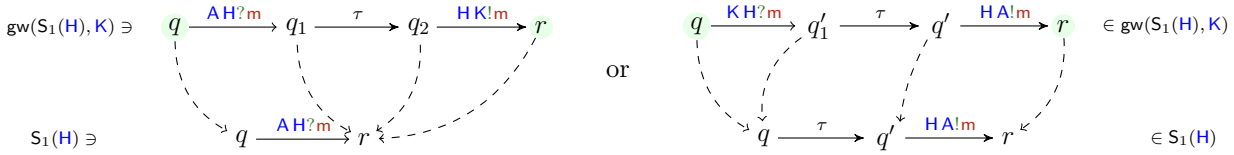
$\boxed{s(\mathsf{H}) = q_2}$ The proof is similar to the previous case except that, instead of Equation (C.4) above. We have $s \xrightarrow{\mathsf{H} \longrightarrow \mathsf{K}:\,\mathsf{m}} s''$.

$\boxed{s(\mathsf{H}) = q_1'}$ The thesis is immediate by the fact that $s \xrightarrow{\tau} s[\mathsf{H} \mapsto q']$ and $\overset{\mathsf{H}}{\mathsf{K}}\rfloor s[\mathsf{H} \mapsto q'] = \overset{\mathsf{H}}{\mathsf{K}}\rfloor s$ since

$$\overset{\mathsf{H}}{\mathsf{K}}\rfloor s(\mathsf{H}) = q, \qquad \text{and} \qquad \bar{s} = (\overset{\mathsf{H}}{\mathsf{K}}\rfloor s)[\mathsf{H} \mapsto q']$$

$\boxed{s(\mathsf{H}) = q'}$ The thesis follows vacuously because $\overset{\mathsf{H}}{\mathsf{K}}\rfloor s(\mathsf{H}) = q'$ hence $\overset{\mathsf{H}}{\mathsf{K}}\rfloor s$ cannot have τ-transitions involving H.

Case $s(\mathsf{H})$ external By our gateway construction (cf. Definition 4.1), we have



where the highlighted states on the left (resp. right) are those non-internal on an exporting (resp. importing) segment. Let us consider each possibility.

$\boxed{s(\mathsf{H}) = q \text{ (exporting)}}$ The thesis follows vacuously because $\overset{\mathsf{H}}{\mathsf{K}}\rfloor s(\mathsf{H}) = q$ hence $\overset{\mathsf{H}}{\mathsf{K}}\rfloor s$ cannot have τ-transitions involving H.

$\boxed{s(\mathsf{H}) = q \text{ (importing)}}$ By Proposition 5.9 and the definition of gateway embedding (cf. Definition 5.6) $\Gamma_{\mathsf{H},\mathsf{K}}(\mathsf{S}_1(\mathsf{H}), s(\mathsf{H})) = q \asymp \Gamma_{\mathsf{K},\mathsf{H}}(\mathsf{S}_2(\mathsf{K}), s(\mathsf{K}))$. Therefore, by the sequentiality of K and the lock freedom of S_2 there is a reachable configuration $s'' \in \mathcal{R}([\![\mathsf{S}]\!], s)$ such that

$$s'' \xrightarrow{\mathsf{K} \longrightarrow \mathsf{H} : \mathsf{m}} \hat{s} \xrightarrow{\tau} s'$$

where H is involved in the τ transition and $\overset{\mathsf{H}}{\mathsf{K}}\rfloor \hat{s} = (\overset{\mathsf{H}}{\mathsf{K}}\rfloor s)[\mathsf{H} \mapsto q'] = \bar{s}$.

$\boxed{s(\mathsf{H}) = r \text{ (exporting)}}$ By definition of τ!-CFSMs, there are a participant $\mathsf{B} \neq \mathsf{H}$ of S_1 and a message n such that

$$r \xrightarrow{\tau} r_1 \xrightarrow{\mathsf{H}\,\mathsf{B}!\mathsf{n}} r_2 \qquad \text{are transitions in } \mathsf{S}_1(\mathsf{H})$$

and we have

$$\overset{\mathsf{H}}{\mathsf{K}}\rfloor s(\mathsf{H}) = r, \qquad \text{and} \qquad \bar{s} = (\overset{\mathsf{H}}{\mathsf{K}}\rfloor s)[\mathsf{H} \mapsto r_1] \tag{C.5}$$

Again by our gateway construction,

$$r \xrightarrow{\mathsf{K}\,\mathsf{H}?\mathsf{n}} r_1' \xrightarrow{\tau} r_1 \xrightarrow{\mathsf{H}\,\mathsf{B}!\mathsf{n}} r_2 \qquad \text{are transition in } \mathsf{gw}(\mathsf{S}_1(\mathsf{H}), \mathsf{K}). \tag{C.6}$$

By Proposition 5.9 and the definition of gateway embedding (cf. Definition 5.6)

$$\Gamma_{\mathsf{H},\mathsf{K}}(\mathsf{S}_1(\mathsf{H}), s(\mathsf{H})) = r \asymp \Gamma_{\mathsf{K},\mathsf{H}}(\mathsf{S}_2(\mathsf{K}), s(\mathsf{K}))$$

Therefore, by the sequentiality of K and the lock freedom of S_2 there is a reachable configuration s'' of S such that

$$s'' \xrightarrow{\mathsf{K} \longrightarrow \mathsf{H} : \mathsf{n}} \hat{s} \xrightarrow{\tau} s'$$

where H is involved in the τ transition and $\overset{\mathsf{H}}{\mathsf{K}}\rfloor \hat{s} = (\overset{\mathsf{H}}{\mathsf{K}}\rfloor s)[\mathsf{H} \mapsto r_1'] $. Hence, $\overset{\mathsf{H}}{\mathsf{K}}\rfloor s' = (\overset{\mathsf{H}}{\mathsf{K}}\rfloor s)[\mathsf{H} \mapsto r_1] = \bar{s}$.

$\boxed{s(\mathsf{H}) = r \text{ (importing)}}$ In this case the state $s(\mathsf{H})$ in the gateway $\mathsf{gw}(\mathsf{S}_1(\mathsf{H}), \mathsf{K})$ is the starting state of an importing segment and the proof goes as in the corresponding case above. □

Proposition 8.5 (Weak lifting of communicating systems)**.** *Let S_1 and S_2 be two (H, K)-composable and deadlock-free communicating systems with H and K sequential and let $\mathsf{S} = \mathsf{S}_1 \,\mathsf{H} \!\!\leftrightarrow\!\! \mathsf{K}\, \mathsf{S}_2$. For all $s \in \mathcal{R}([\![\mathsf{S}]\!])$, if $\overset{\mathsf{H}}{\mathsf{K}}\rfloor s \to$ or $s \lfloor \overset{\mathsf{K}}{\mathsf{H}} \to$ then $s \to$.*

Proof. We show that for all λ, $\overset{\mathsf{H}}{\mathsf{K}}\rfloor s \xrightarrow{\lambda}$ in $[\![\mathsf{S}_1]\!]$ implies $s \to$ in $[\![\mathsf{S}]\!]$. (The case of the right-projection is symmetric and therefore omitted.) Note that all participants involved in λ are in S_1.

If H is not involved in λ then $s \xrightarrow{\lambda}$ since $\overset{\mathsf{H}}{\mathsf{K}}\rfloor s(\mathsf{X}) = s(\mathsf{X})$ for all $\mathsf{X} \in \mathsf{ptp}(\lambda)$ by left-projection (cf. Definition 5.1). Otherwise, H is either the sender or the receiver of λ. Recall that exporting and importing

segments correspond to input and output transitions of the interface machine H; pictorially

where the fresh states are highlighted and dashed arrows represent the left-projection mapping.

$\boxed{s(\mathsf{H}) = q.}$ Since $s(\mathsf{H}) =^{\mathsf{H}}_{\mathsf{K}}\lfloor s(\mathsf{H})$ then $s \xrightarrow{\lambda}$ because $s(\mathsf{A}) =^{\mathsf{H}}_{\mathsf{K}}\lfloor s(\mathsf{A})$ by Definition 5.1.

$\boxed{s(\mathsf{H}) = q'.}$ By Proposition 5.9, $\Gamma_{\mathsf{H},\mathsf{K}}(\mathsf{S}_1(\mathsf{H}), s(\mathsf{H})) = q \asymp \Gamma_{\mathsf{K},\mathsf{H}}(\mathsf{S}_2(\mathsf{K}), s(\mathsf{K}))$ therefore the current state $\mathsf{S}(\mathsf{K})$ of the gateway $\mathsf{gw}(\mathsf{S}_2(\mathsf{K}), \mathsf{H})$ is the starting state of an importing segment with an enabled input transition of message m from K. Hence, $s \xrightarrow{\mathsf{H} \longrightarrow \mathsf{K}:\, \mathsf{m}}$.

$\boxed{s(\mathsf{H}) = r.}$ The proof is like in the cases $s(\mathsf{H}) = q$ or $s(\mathsf{H}) = \hat{q}$ since r is the starting state of an exporting or exporting segment of the gateway $\mathsf{gw}(\mathsf{S}_1(\mathsf{H}), \mathsf{K})$.

$\boxed{s(\mathsf{H}) = \hat{q}.}$ By Proposition 5.9, $\Gamma_{\mathsf{H},\mathsf{K}}(\mathsf{S}_1(\mathsf{H}), s(\mathsf{H})) = \hat{q} \asymp \Gamma_{\mathsf{K},\mathsf{H}}(\mathsf{S}_2(\mathsf{K}), s(\mathsf{K}))$ therefore the current state $\mathsf{S}(\mathsf{K})$ of the gateway $\mathsf{gw}(\mathsf{S}_2(\mathsf{K}), \mathsf{H})$ is in a state $s(\mathsf{K})$, with enabled output transitions to H. Note that $s(\mathsf{K})$ is also a state of the interface CFSM $\mathsf{S}_2(\mathsf{K})$ (since $s(\mathsf{K}) = s\lfloor^{\mathsf{K}}_{\mathsf{H}}(\mathsf{K})$) and that for each output transition from $s(\mathsf{K})$ in the gateway there is a corresponding input transition from $s(\mathsf{K})$ in the CFSM interface $\mathsf{S}_2(\mathsf{K})$. By compatibility, one of such inputs should match an output from the state $\hat{q}$ of the interface CFSM $\mathsf{S}_1(\mathsf{H})$. Correspondingly, an input from $\hat{q}$ in the gateway $\mathsf{gw}(\mathsf{S}_1(\mathsf{H}), \mathsf{K})$ matches an output of $s(\mathsf{K})$ in $\mathsf{gw}(\mathsf{S}_2(\mathsf{K}), \mathsf{H})$. By Definition 3.3, $s \xrightarrow{\mathsf{K} \longrightarrow \mathsf{H}:\, \mathsf{n}}$ for some message n.

$\boxed{s(\mathsf{H}) = \hat{q}'.}$ We have $s \xrightarrow{\lambda}$ because $s(\mathsf{A}) =^{\mathsf{H}}_{\mathsf{K}}\lfloor s(\mathsf{A})$ by Definition 5.1.

$\boxed{s(\mathsf{H}) = \hat{r}.}$ The proof is like in the cases $s(\mathsf{H}) = q$ or $s(\mathsf{H}) = \hat{q}$ since r is the starting state of an exporting or exporting segment of the gateway $\mathsf{gw}(\mathsf{S}_1(\mathsf{H}), \mathsf{K})$. □

Proposition 8.6 (Reflecting interactions on sub-systems)**.** *Let* $\mathsf{S} = \mathsf{S}_1 {}^{\mathsf{H}}\!\!\not\leftrightarrow^{\mathsf{K}} \mathsf{S}_2$ *be the composition of* S_1 *and* S_2*, two* (H, K)*-composable communicating systems or* τ*!-systems. For all* $s \in \mathcal{R}(\llbracket \mathsf{S} \rrbracket)$ *and* $s \xrightarrow{\lambda} s'$*,*

i) *if* $\lambda = \tau$ *then* $(^{\mathsf{H}}_{\mathsf{K}}\lfloor s \xrightarrow{\tau} {}^{\mathsf{H}}_{\mathsf{K}}\lfloor s'$ *or* $^{\mathsf{H}}_{\mathsf{K}}\lfloor s =^{\mathsf{H}}_{\mathsf{K}}\lfloor s')$

ii) *if* $\lambda \neq \tau$ *then* $^{\mathsf{H}}_{\mathsf{K}}\lfloor s \xrightarrow{\lambda} {}^{\mathsf{H}}_{\mathsf{K}}\lfloor s'$*, or* $^{\mathsf{H}}_{\mathsf{K}}\lfloor s \xrightarrow{\tau} \xrightarrow{\lambda} {}^{\mathsf{H}}_{\mathsf{K}}\lfloor s'$*, or else* $^{\mathsf{H}}_{\mathsf{K}}\lfloor s =^{\mathsf{H}}_{\mathsf{K}}\lfloor s'$*.*

The same holds for the $\_\lfloor^{\mathsf{K}}_{\mathsf{H}}$ *projection.*

Proof. $\boxed{\lambda = \tau}$. Implication (i) applies only to τ!-systems. Either H is not involved in the τ-transition or it is. In the former case the transition occurs because, for a participant $\mathsf{A} \neq \mathsf{H}$, $s(\mathsf{A}) \xrightarrow{\tau}$ hence we get $^{\mathsf{H}}_{\mathsf{K}}\lfloor s \xrightarrow{\tau} {}^{\mathsf{H}}_{\mathsf{K}}\lfloor s'$, by definition of projection.

If H is involved in the τ-transition, by definition of gateway (cf. Definition 4.1) and left-projection (cf. Definition 5.1), we have that

- $q \xrightarrow{\mathsf{A}\,\mathsf{H}?\mathsf{m}} s(\mathsf{H}) \xrightarrow{\tau} s'(\mathsf{H}) \xrightarrow{\mathsf{H}\,\mathsf{K}!\mathsf{m}} r$
- $^{\mathsf{H}}_{\mathsf{K}}\lfloor s(\mathsf{H}) \xrightarrow{\mathsf{K}\,\mathsf{H}?\mathsf{m}} s(\mathsf{H}) \xrightarrow{\tau} s'(\mathsf{H}) \xrightarrow{\mathsf{H}\,\mathsf{A}!\mathsf{m}} r$

are the only possible sequences of transitions in the gateway $\mathsf{S}(\mathsf{H})$ for some q, r which are also states of the interface participant H.

In the first case $\mathsf{H}_{\mathsf{K}}\rfloor s = {}_{\mathsf{K}}^{\mathsf{H}}\rfloor s'$ because for any $\mathsf{X} \neq \mathsf{H} \in \mathsf{S}_1$, $s(\mathsf{X}) = s'(\mathsf{X})$. Also, by definition of projection, ${}_{\mathsf{K}}^{\mathsf{H}}\rfloor s'(\mathsf{H}) = {}_{\mathsf{K}}^{\mathsf{H}}\rfloor s(\mathsf{H})$. In the second case we have ${}_{\mathsf{K}}^{\mathsf{H}}\rfloor s'(\mathsf{H}) = s'(\mathsf{H})$ which implies ${}_{\mathsf{K}}^{\mathsf{H}}\rfloor s(\mathsf{H}) \xrightarrow{\tau} {}_{\mathsf{K}}^{\mathsf{H}}\rfloor s'(\mathsf{H}) \in \mathsf{S}_1(\mathsf{H})$ which, by the synchronous semantics given in Definition 3.3, yields ${}_{\mathsf{K}}^{\mathsf{H}}\rfloor s \xrightarrow{\tau} {}_{\mathsf{K}}^{\mathsf{H}}\rfloor s'$.

$\boxed{\lambda \neq \tau}$ Assume $\lambda = \mathsf{Y}{\to}\mathsf{X}\colon \mathsf{m}$. We have the following cases:

1. neither X nor Y are an interface,
2. only one between X and Y is an interface,
3. both X and Y are an interface.

In case (1) by Definition 5.1, we immediately get ${}_{\mathsf{K}}^{\mathsf{H}}\rfloor s \xrightarrow{\lambda} {}_{\mathsf{K}}^{\mathsf{H}}\rfloor s'$ since the projection operation leaves unchanged the states of non-interface participants. In the other cases we can assume without loss of generality that $\mathsf{Y} = \mathsf{H}$; the cases where $\mathsf{Y} = \mathsf{K}$ are symmetric and therefore omitted.

In case (2), by our gateway construction and by the left-projection operation we have

$$
\begin{aligned}
{}_{\mathsf{K}}^{\mathsf{H}}\rfloor s(\mathsf{H}) \xrightarrow{\mathsf{K}\,\mathsf{H}?\mathsf{m}} \xrightarrow{\tau} s(\mathsf{H}) \xrightarrow{\mathsf{H}\,\mathsf{X}!\mathsf{m}} s'(\mathsf{H}) = {}_{\mathsf{K}}^{\mathsf{H}}\rfloor s'(\mathsf{H}) \in \mathsf{S}(\mathsf{H}) && \text{for } \tau!\text{-systems} \\
{}_{\mathsf{K}}^{\mathsf{H}}\rfloor s(\mathsf{H}) \xrightarrow{\mathsf{K}\,\mathsf{H}?\mathsf{m}} \xrightarrow{\mathsf{H}\,\mathsf{X}!\mathsf{m}} s'(\mathsf{H}) = {}_{\mathsf{K}}^{\mathsf{H}}\rfloor s'(\mathsf{H}) \in \mathsf{S}(\mathsf{H}) && \text{for communicating systems}
\end{aligned}
$$

respectively obtained from ${}_{\mathsf{K}}^{\mathsf{H}}\rfloor s(\mathsf{H}) \xrightarrow{\tau} s(\mathsf{H}) \xrightarrow{\mathsf{H}\,\mathsf{X}!\mathsf{m}} {}_{\mathsf{K}}^{\mathsf{H}}\rfloor s'(\mathsf{H}) \in \mathsf{S}_1(\mathsf{H})$ or ${}_{\mathsf{K}}^{\mathsf{H}}\rfloor s(\mathsf{H}) = s(\mathsf{H}) \xrightarrow{\mathsf{H}\,\mathsf{X}!\mathsf{m}} {}_{\mathsf{K}}^{\mathsf{H}}\rfloor s'(\mathsf{H}) \in \mathsf{S}_1(\mathsf{H})$. Hence we get ${}_{\mathsf{K}}^{\mathsf{H}}\rfloor s \xrightarrow{\tau} \xrightarrow{\mathsf{H}\to\mathsf{X}\colon \mathsf{m}} {}_{\mathsf{K}}^{\mathsf{H}}\rfloor s'$ in the former case and ${}_{\mathsf{K}}^{\mathsf{H}}\rfloor s \xrightarrow{\mathsf{H}\to\mathsf{X}\colon \mathsf{m}} {}_{\mathsf{K}}^{\mathsf{H}}\rfloor s'$ in the latter case.

In case (3) we have that, for a state $q \in \mathsf{S}_1(\mathsf{H})$,

$$
\begin{aligned}
q \xrightarrow{\mathsf{X}\,\mathsf{H}?\mathsf{m}} \xrightarrow{\tau} s(\mathsf{H}) \xrightarrow{\mathsf{H}\,\mathsf{K}!\mathsf{m}} s'(\mathsf{H}) = {}_{\mathsf{K}}^{\mathsf{H}}\rfloor s(\mathsf{H}) = {}_{\mathsf{K}}^{\mathsf{H}}\rfloor s'(\mathsf{H}) \in s(\mathsf{H}) && \text{for } \tau!\text{-systems} \\
q \xrightarrow{\mathsf{X}\,\mathsf{H}?\mathsf{m}} s(\mathsf{H}) \xrightarrow{\mathsf{H}\,\mathsf{K}!\mathsf{m}} s'(\mathsf{H}) = {}_{\mathsf{K}}^{\mathsf{H}}\rfloor s(\mathsf{H}) = {}_{\mathsf{K}}^{\mathsf{H}}\rfloor s'(\mathsf{H}) \in s(\mathsf{H}) && \text{for communicating systems}
\end{aligned}
$$

Hence, in either case we get $s\lfloor_{\mathsf{H}}^{\mathsf{K}} = s'\lfloor_{\mathsf{H}}^{\mathsf{K}}$ because the projection operation leaves unchanged the states of participants $\mathsf{X} \neq \mathsf{H} \in \mathsf{S}_1$. □