Safe Composition of Systems of Communicating Finite State Machines

Franco Barbanera[1], *Rolf Hennicker* [2],

[1]University of Catania (IT) [2]LMU Munich (D)

ICE - June 20-22, 2024, Groeningen (NL)

OVERVIEW

- Introduction: the need for system composability
- The "participants-as-interfaces" (PaI) approach to (binary) system composition
- ICE'24: PaI for multicomposition (for CFSM systems).

OVERVIEW

- Introduction: the need for system composability
- The "participants-as-interfaces" (PaI) approach to (binary) system composition
- ICE'24: PaI for multicomposition (for CFSM systems).

OVERVIEW

- Introduction: the need for system composability
- The "participants-as-interfaces" (PaI) approach to (binary) system composition
- ICE'24: PaI for multicomposition (for CFSM systems).

OVERVIEW

- Introduction: the need for system composability
- The "participants-as-interfaces" (PaI) approach to (binary) system composition
- ICE'24: PaI for multicomposition (for CFSM systems).

The need of systems composability

- Concurrent/Distributed systems are

  not STAND-ALONE ENTITIES

- (expecially nowadays) they are parts of

  JIGSAWS NEVER COMPLETELY TERMINATED

The need of systems composability

- Concurrent/Distributed systems are

not STAND-ALONE ENTITIES

- (expecially nowadays) they are parts of

JIGSAWS NEVER COMPLETELY TERMINATED

The need of systems composability

- Concurrent/Distributed systems are

  not STAND-ALONE ENTITIES

- (expecially nowadays) they are parts of

  JIGSAWS NEVER COMPLETELY TERMINATED

The need of systems composability

- Concurrent/Distributed systems are

  not STAND-ALONE ENTITIES

- (expecially nowadays) they are parts of

  JIGSAWS NEVER COMPLETELY TERMINATED

Good composition methods

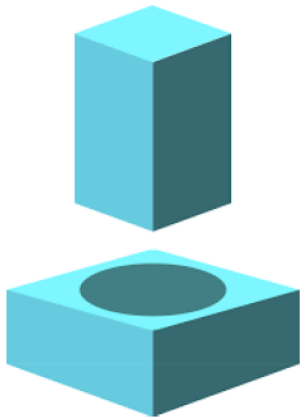
They should be

Good composition methods

They should be

- CONSERVATIVE

  Altering as less as possible the single systems



Good composition methods

They should be

- CONSERVATIVE
- FLEXIBLE

i.e. "system independent": the composition mechanism
- is not part of the system
-allows to consider **any** system as potentially **open**

Good composition methods

They should be

- CONSERVATIVE
- FLEXIBLE
- SAFE

  Guaranteeing not to “break” relevant properties of the single systems we compose.

Good composition methods are **safe** when...

If one starts from systems like this....

Good composition methods are **safe** when...

If one starts from systems like this....

Good composition methods are **safe** when...

If one starts from systems like this....

Good composition methods are **safe**

...one does not end up with something like that

Good composition methods are **safe**

...one does not end up with something like that

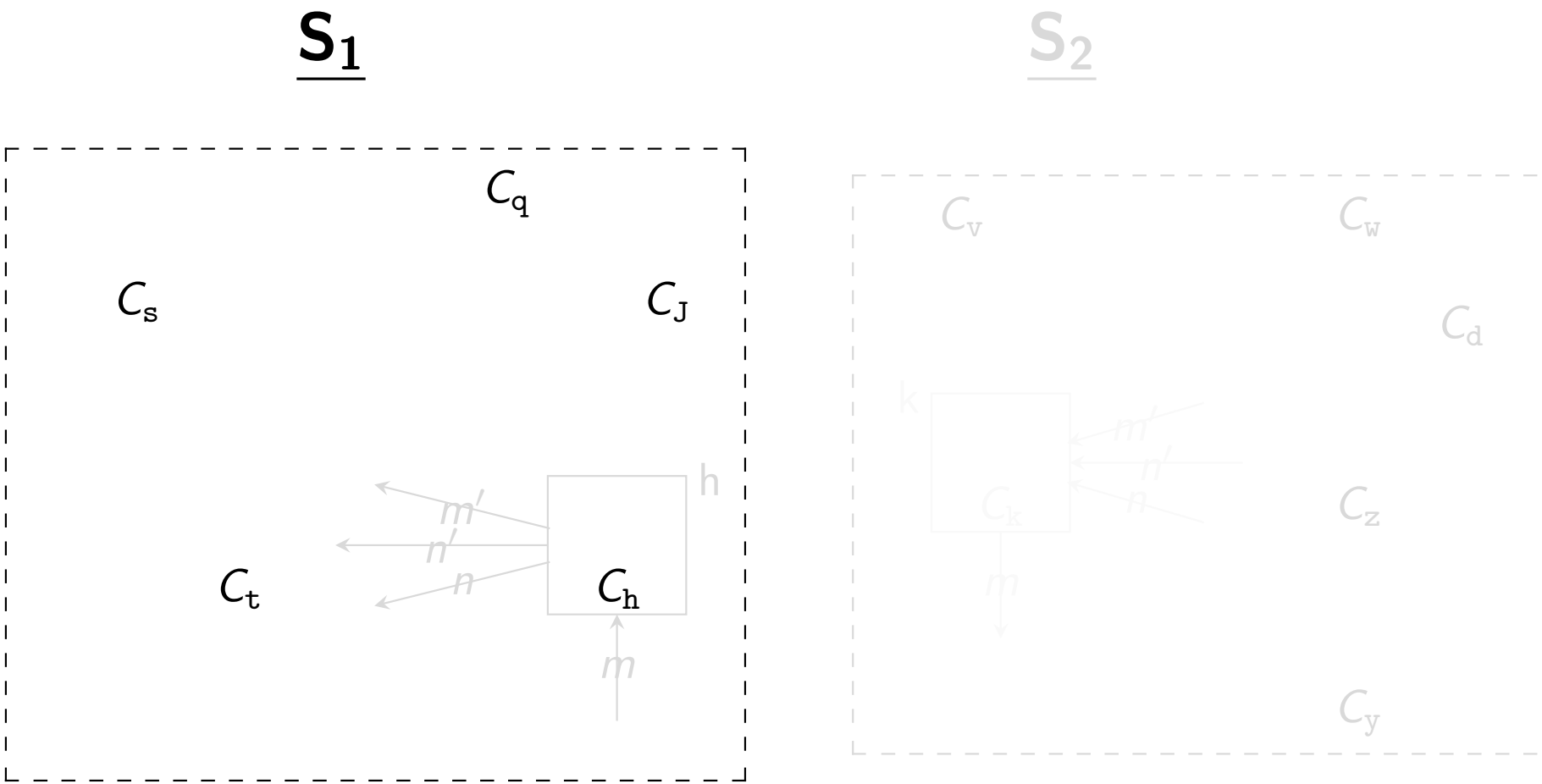
The "participants-as-interfaces" (PaI) approach

For systems with message-passing interactions

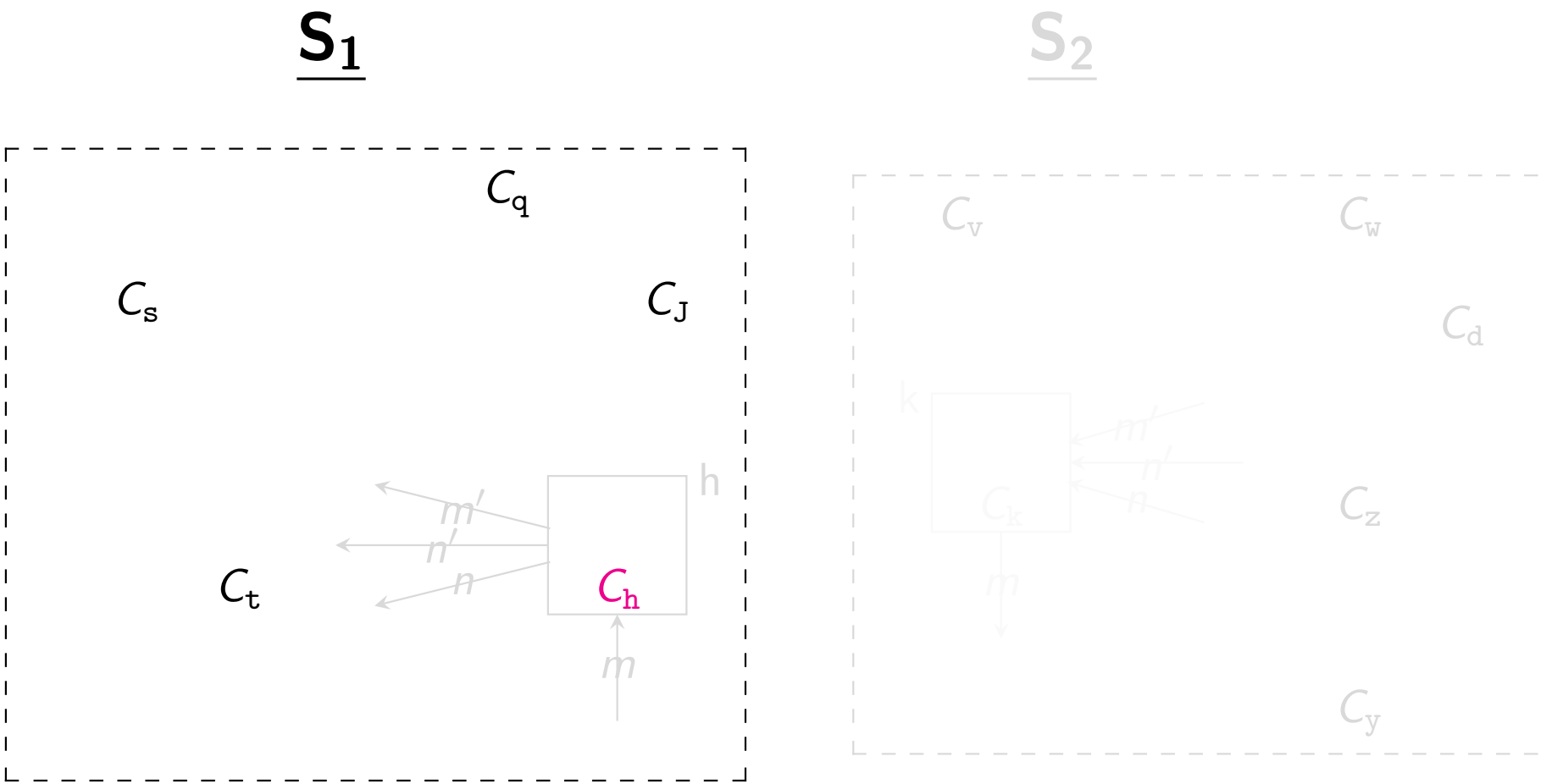
Introduced (as far as we know) in

Franco Barbanera, Ugo de'Liguoro, Rolf Hennicker:
Global Types for Open Systems. **ICE 2018**

The "participants-as-interfaces" (PaI) approach

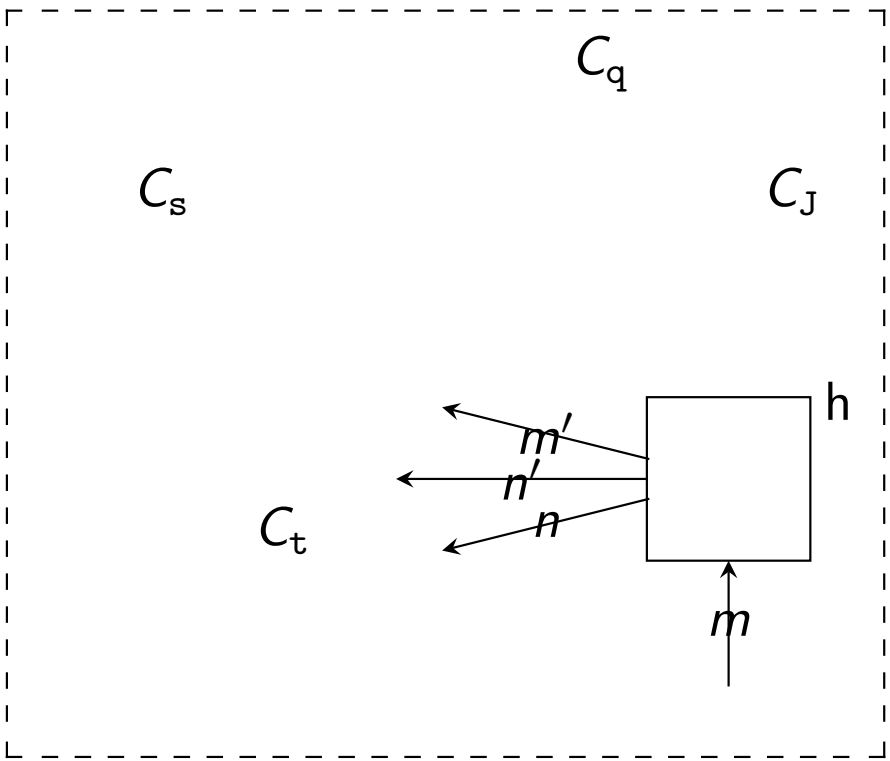


The "participants-as-interfaces" (PaI) approach



The "participants-as-interfaces" (PaI) approach

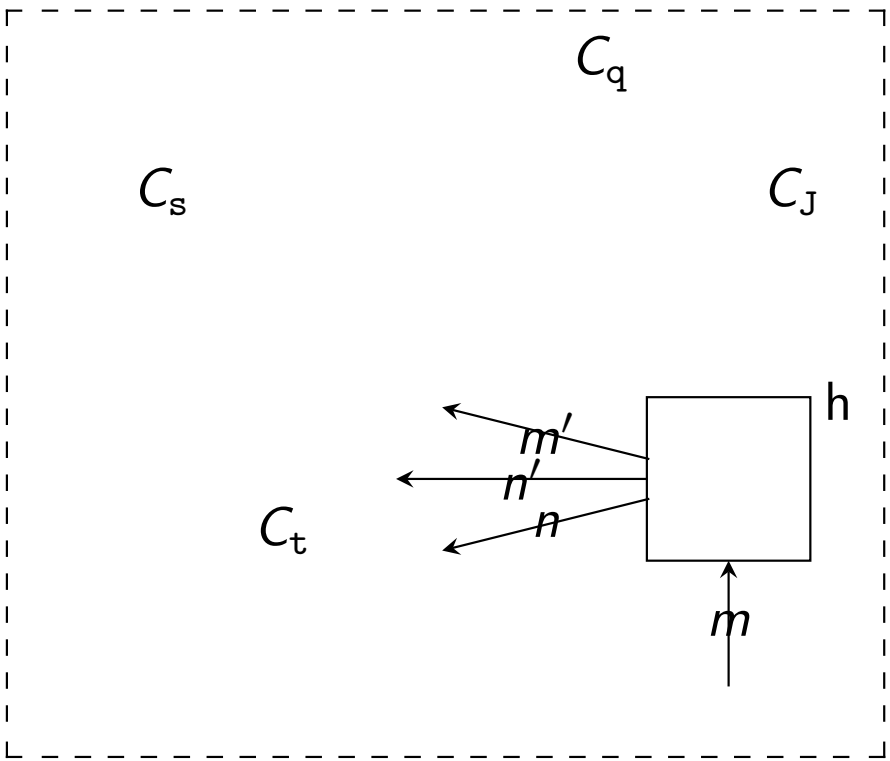
S_1



S_2

The "participants-as-interfaces" (PaI) approach

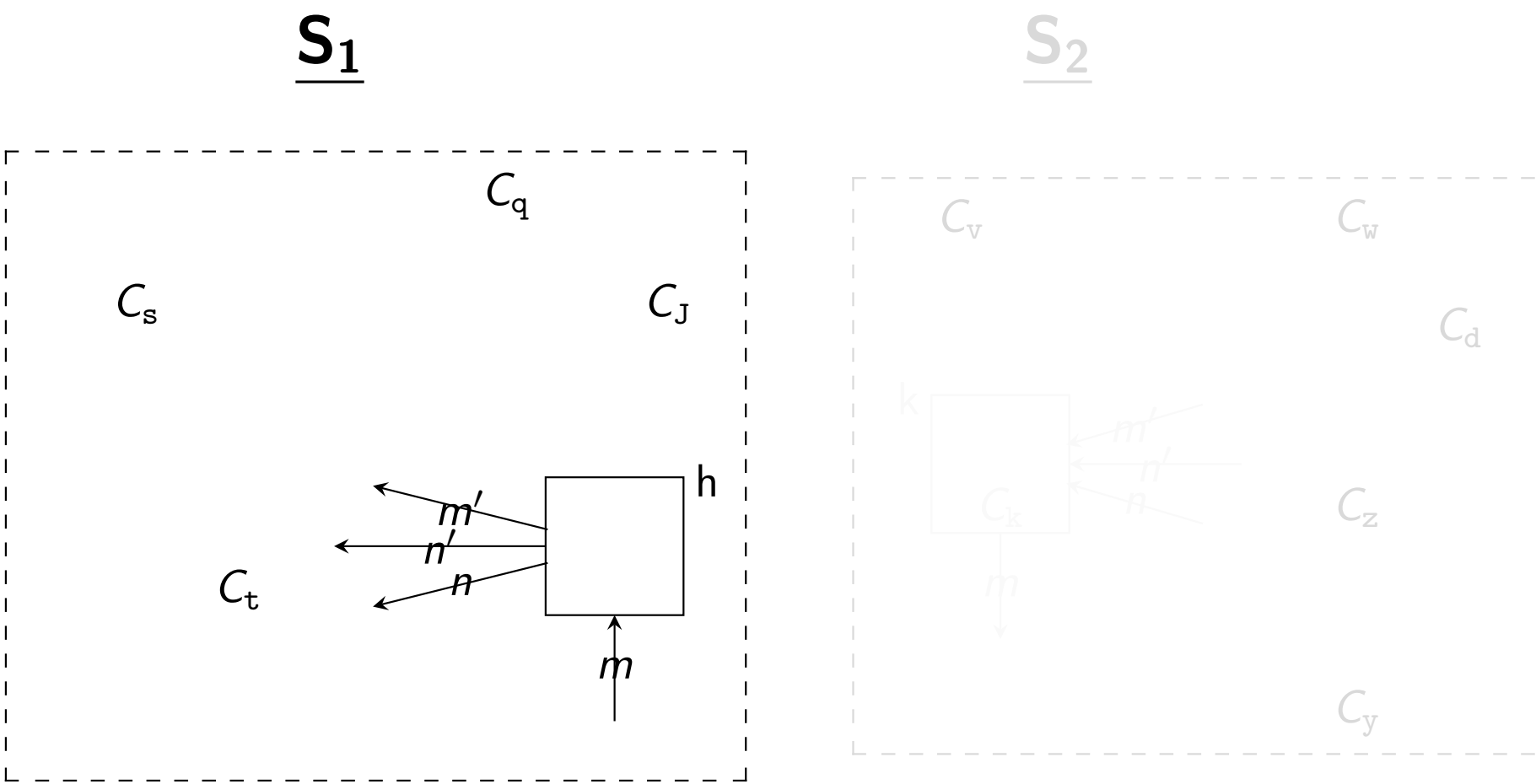
S_1



S_2

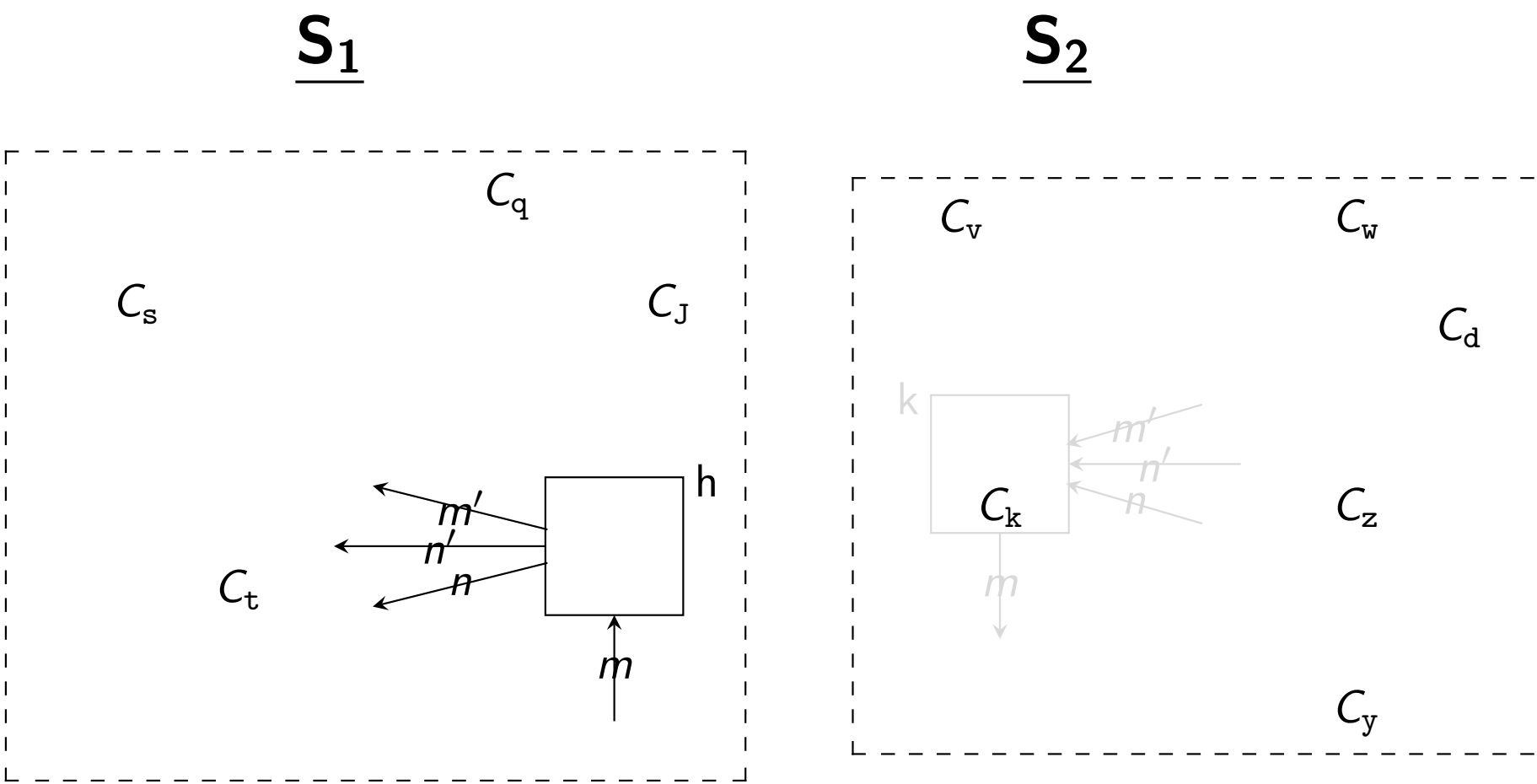
We abstract here from the way communications are performed and from the logical order of the exchanged messages.

The "participants-as-interfaces" (PaI) approach

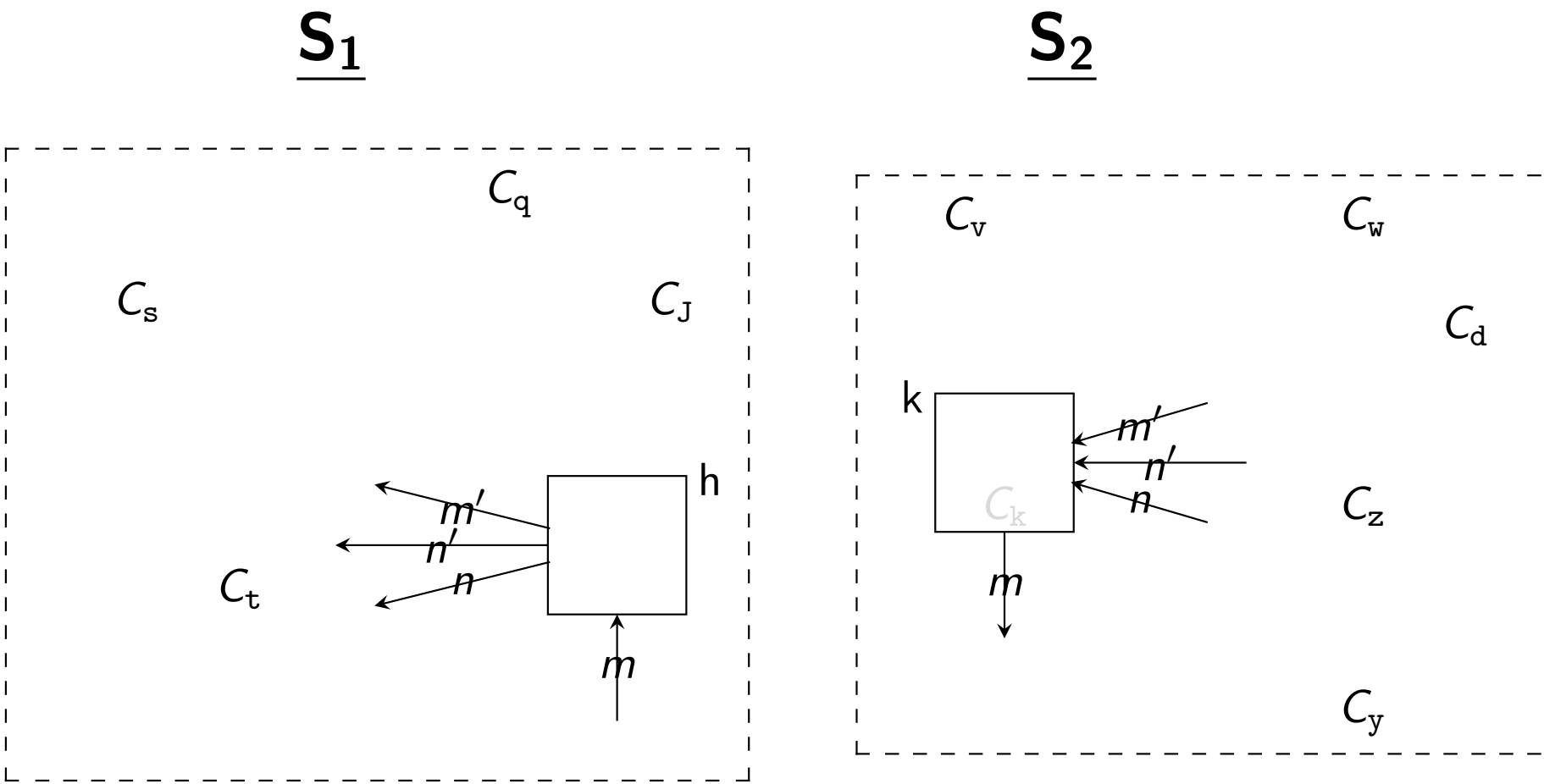


C_h's behaviour can be looked at as an interface (i.e. a description of what can be offered by an outer system)

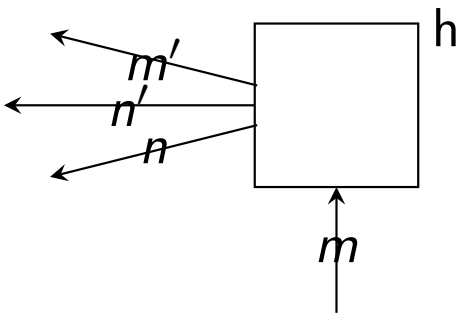
The "participants-as-interfaces" (PaI) approach

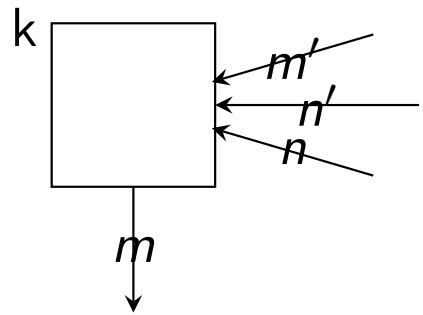


The "participants-as-interfaces" (PaI) approach

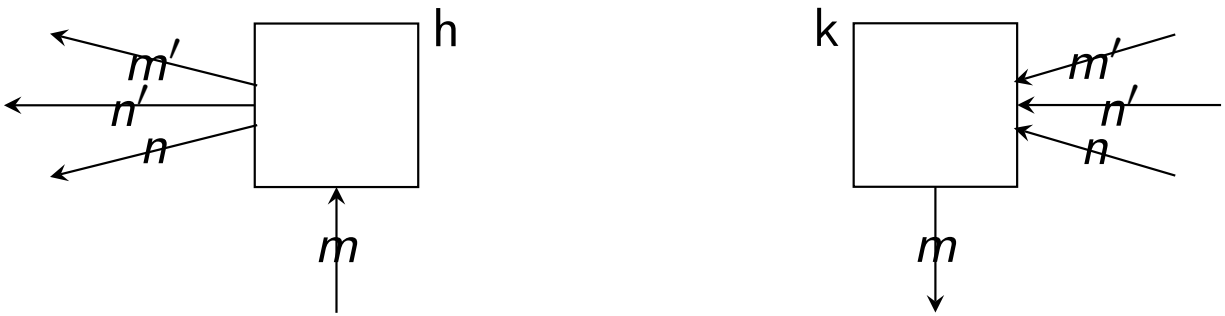


The "participants-as-interfaces" (PaI) approach



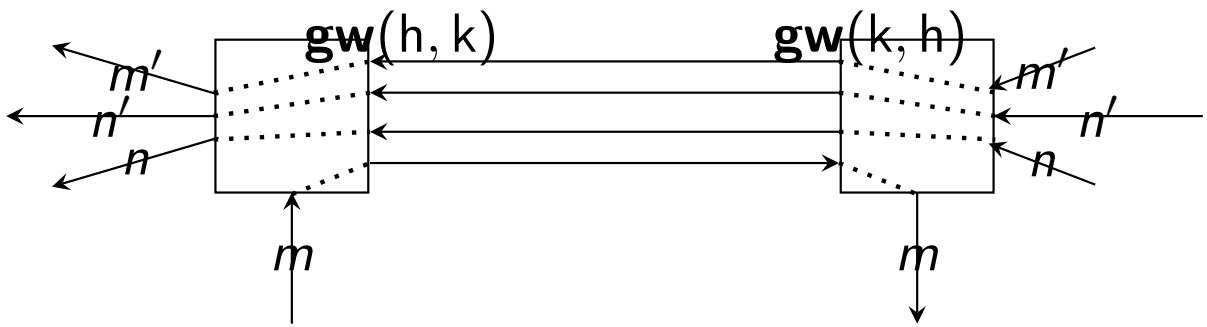


The "participants-as-interfaces" (PaI) approach



COMPATIBLE: an h's input is a k's output, and vice versa

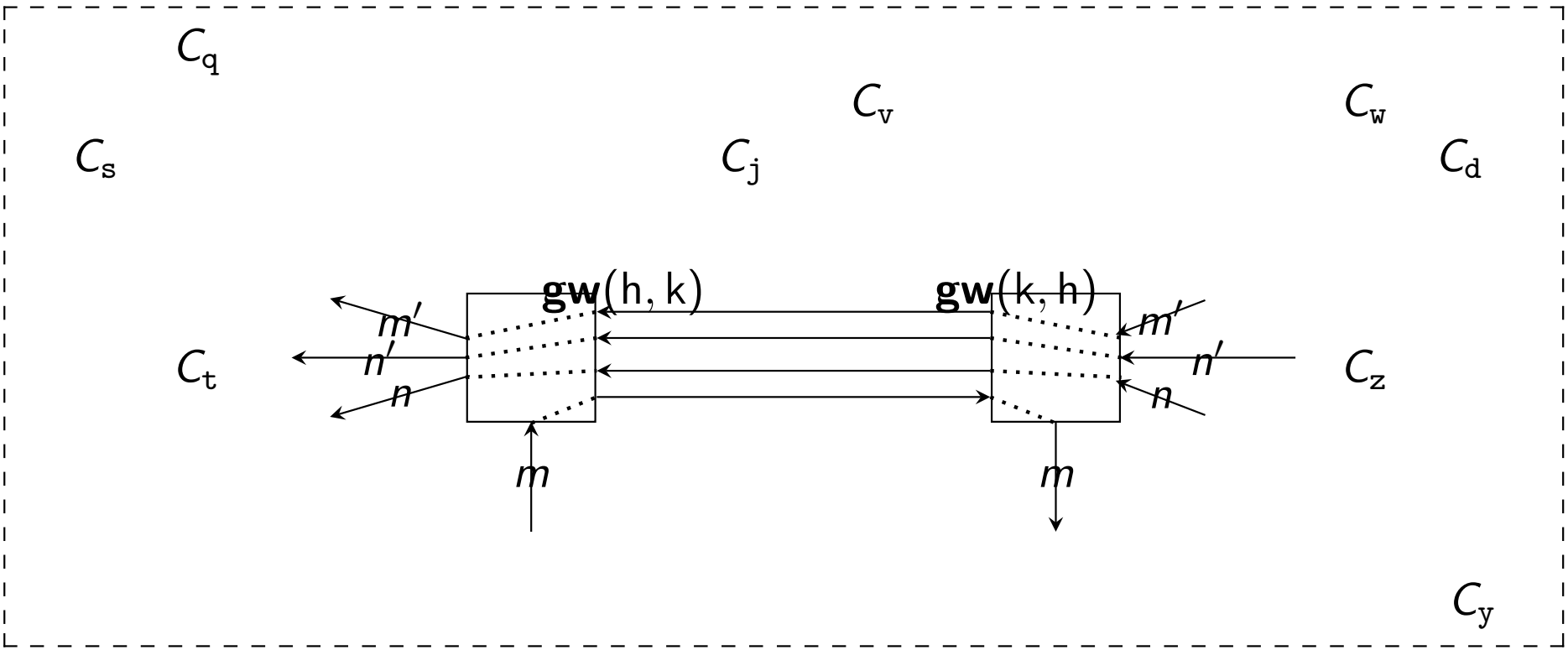
The "participants-as-interfaces" (PaI) approach



Composition via gateways (forwarders)

The "participants-as-interfaces" (PaI) approach

$$\underline{\mathbf{S_1}}^{\mathsf{h}\leftrightarrow\mathsf{k}}\underline{\mathbf{S_2}}$$



The "participants-as-interfaces" (PaI) approach

- CONSERVATIVE ✓
- FLEXIBLE ✓

The "participants-as-interfaces" (PaI) approach

- CONSERVATIVE ✓
- FLEXIBLE ✓

The "participants-as-interfaces" (PaI) approach

- CONSERVATIVE ✓
- FLEXIBLE ✓

The "participants-as-interfaces" (PaI) approach

- SAFE ?

The "participants-as-interfaces" (PaI) approach

- SAFE ?

Which formalism for concurrent system description?

Which communication model?

The "participants-as-interfaces" (PaI) approach

- SAFE ✓

A number of results for PaI binary composition
(see paper's References)

for MPST - MultiParty Session Types - (synchronous)

for CFSM - Communicating Finite State Machines - (synchronous and asynchronous).

The "participants-as-interfaces" (PaI) approach

- SAFE ✓

A number of results for PaI binary composition
(see paper's References)

for MPST - MultiParty Session Types - (synchronous)

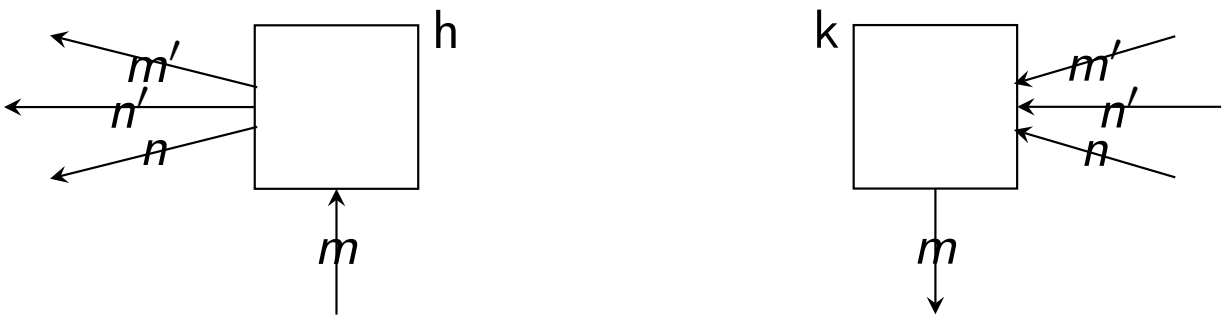
for CFSM - Communicating Finite State Machines -
(synchronous and asynchronous).

Strict conditions to get safeness in synchronous CFSM systems

F. Barbanera, Ivan Lanese, Emilio Tuosto:
Composition of synchronous communicating systems. JLAMP (2023)

The "participants-as-interfaces" (PaI) approach

SAFENESS guaranteed by
Compatibility of Interfaces



COMPATIBILITY $=_{\textbf{roughly}}$ Bisimulation

The "participants-as-interfaces" (PaI) approach

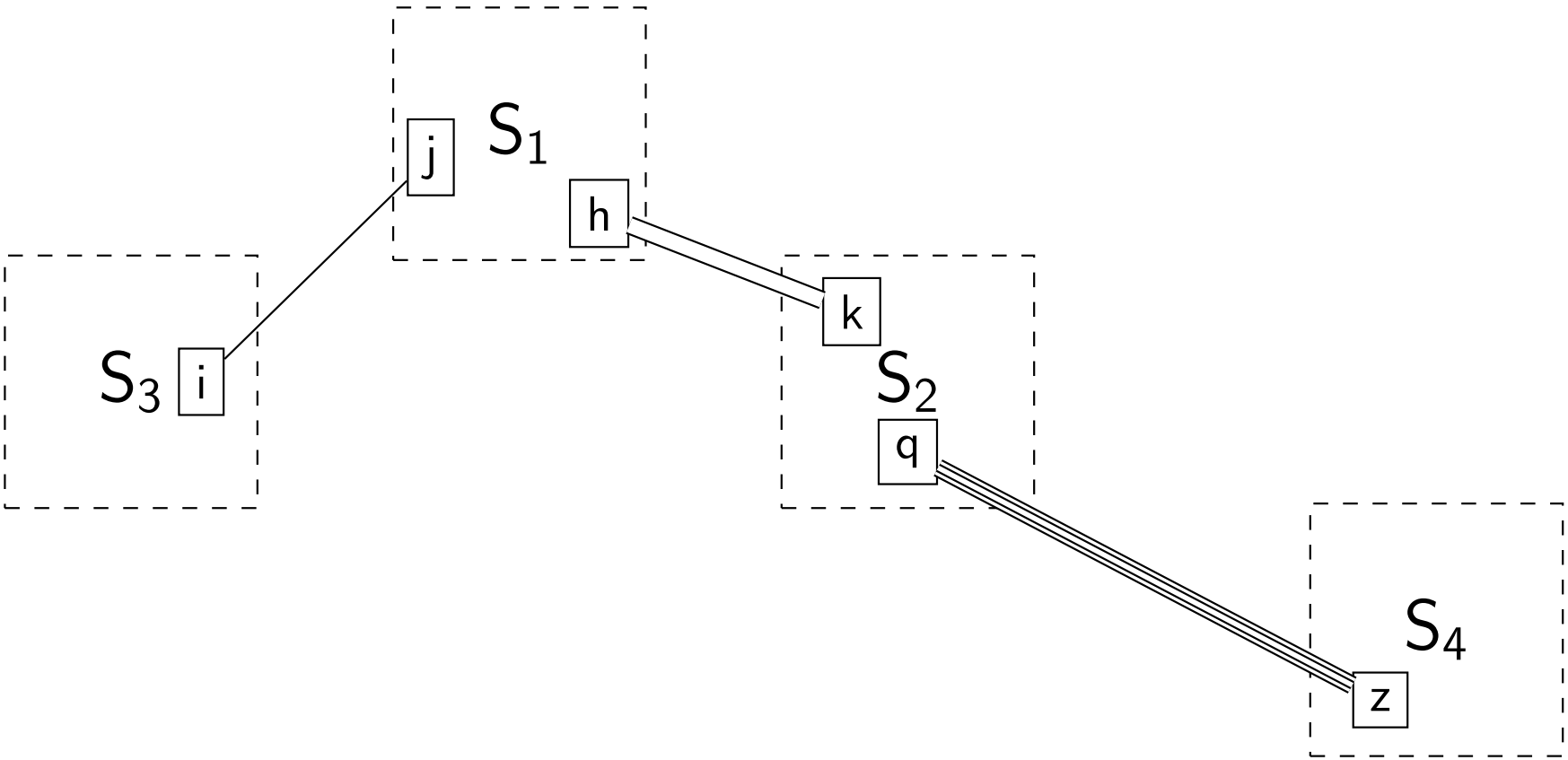
Drawback of binary composition:

By connecting systems two-by-two we get only tree-topologies.

The "participants-as-interfaces" (PaI) approach

Drawback of binary composition:

By connecting systems two-by-two we get only tree-topologies.

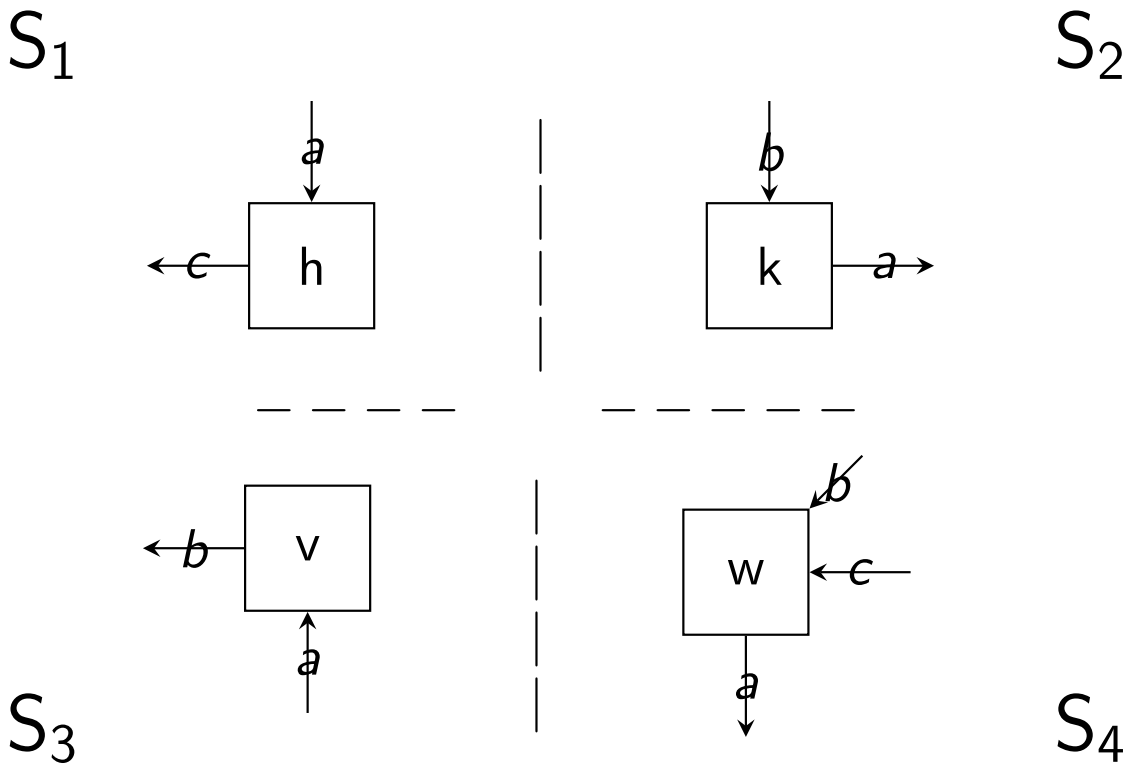


PaI Multicomposition

The composition via gateways trivially extends to simultaneous multiple system composition.

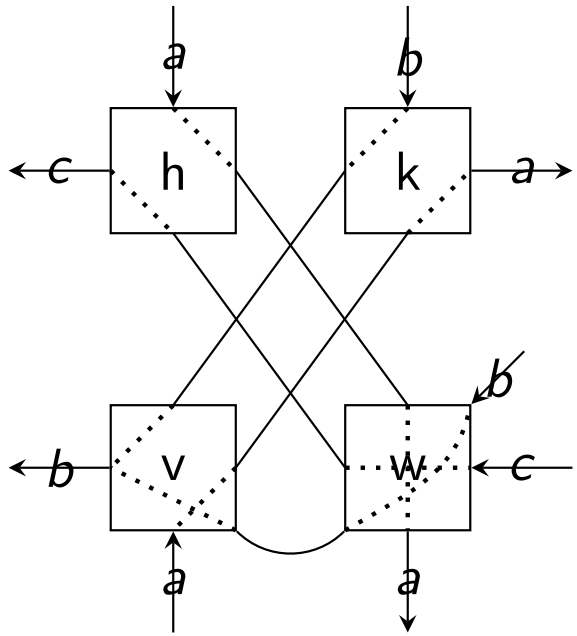
PaI Multicomposition

The composition via gateways trivially extends to simultaneous multiple system composition.



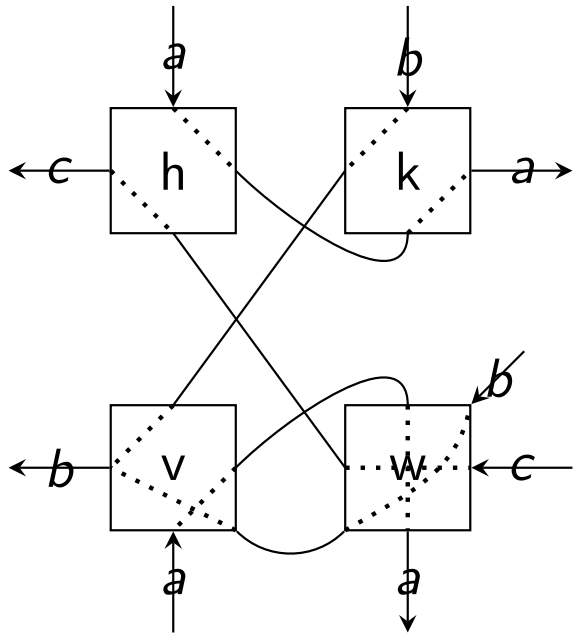
PaI Multicomposition

The composition via gateways trivially extends to simultaneous multiple system composition.



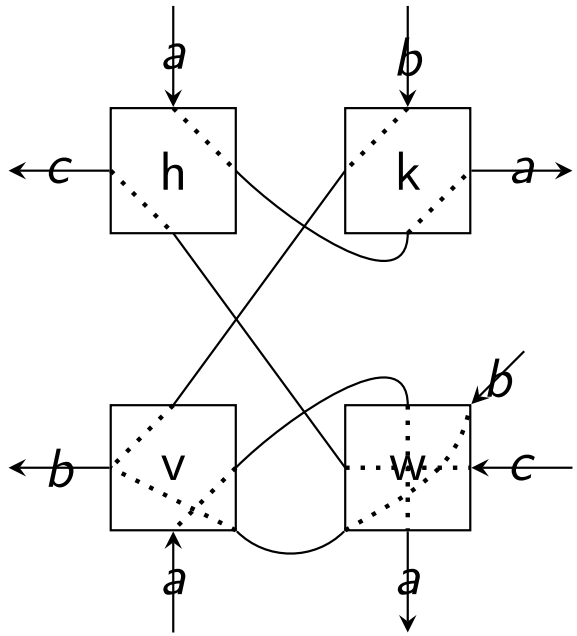
PaI Multicomposition

The composition via gateways trivially extends to simultaneous multiple system composition.



PaI Multicomposition

The composition via gateways trivially extends to simultaneous multiple system composition.



Issues: • Many different "connection policies" (all safe?).

Exploiting PaI multicomposition for (synchronous) MPST

F. Barbanera, M. Dezani-Ciancaglini, L. Gheri, N. Yoshida: Multicompatibility for Multiparty-Session Composition. **PPDP 2023**

Exploiting PaI multicomposition for systems of (asynchronous) CFSMs

Communicating Finite State Machines (CFSMs)

A formalism for the description and the analysis of distributed systems.

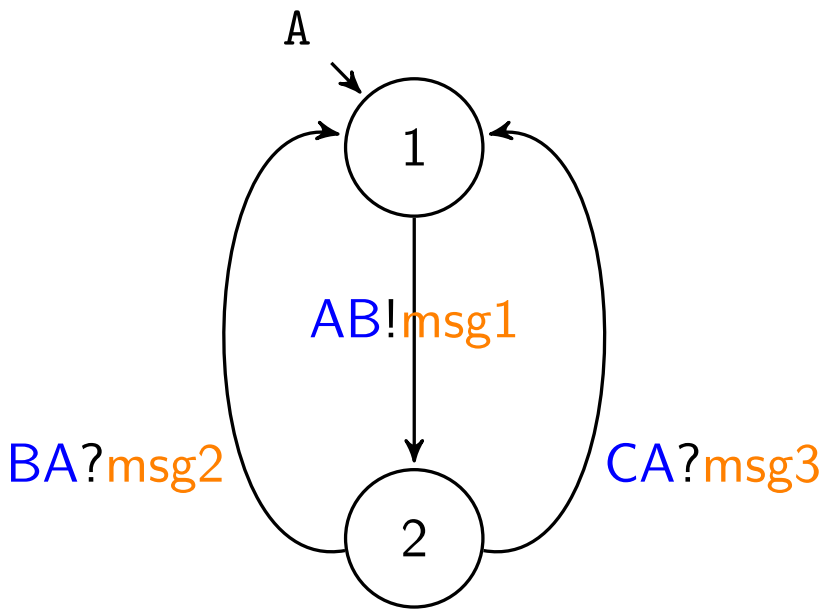
A machine M_A

- M_A can send msg1 to machine M_B;
  asynchronously; *through the directed buffered FIFO channel* AB
- Then, either msg2 or msg3 can be received from M_B or M_C;
  through channels BA or CA;
- and so on....

Communicating Finite State Machines (CFSMs)

A formalism for the description and the analysis of distributed systems.

A machine $M_{\mathtt{A}}$

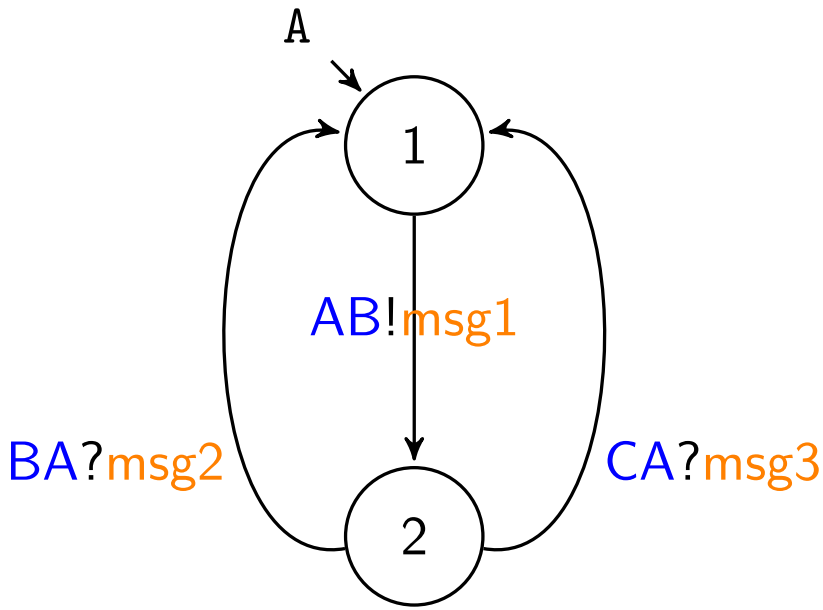


- $M_{\mathtt{A}}$ can send msg1 to machine $M_{\mathtt{B}}$;
  asynchronously; *through the directed buffered FIFO channel* AB
- Then, either msg2 or msg3 can be received from $M_{\mathtt{B}}$ or $M_{\mathtt{C}}$;
  through channels BA or CA;
- and so on....

Communicating Finite State Machines (CFSMs)

A formalism for the description and the analysis of distributed systems.

A machine $M_{\mathtt{A}}$

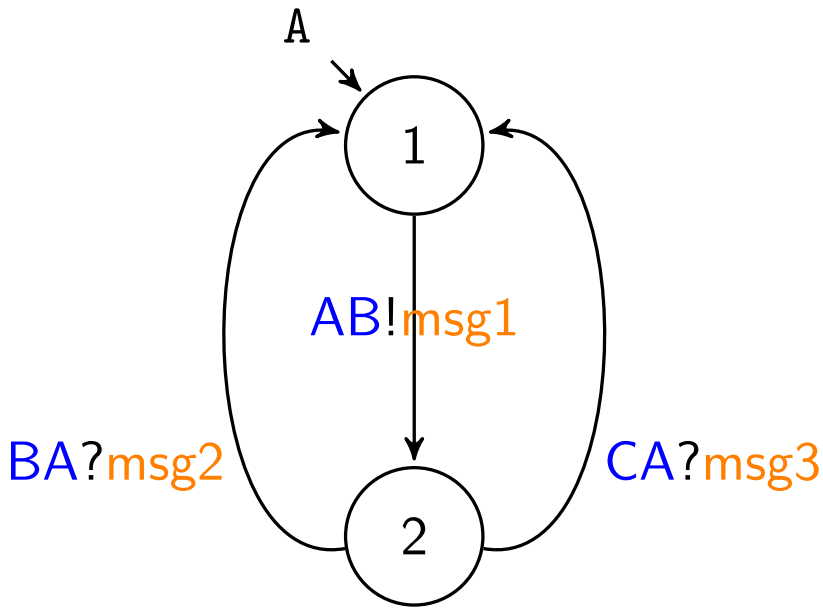


- $M_{\mathtt{A}}$ can send msg1 to machine $M_{\mathtt{B}}$;
  asynchronously; *through the directed buffered FIFO channel* AB
- Then, either msg2 or msg3 can be received from $M_{\mathtt{B}}$ or $M_{\mathtt{C}}$;
  through channels BA or CA;
- and so on....

Communicating Finite State Machines (CFSMs)

A formalism for the description and the analysis of distributed systems.

A machine $M_{\mathtt{A}}$

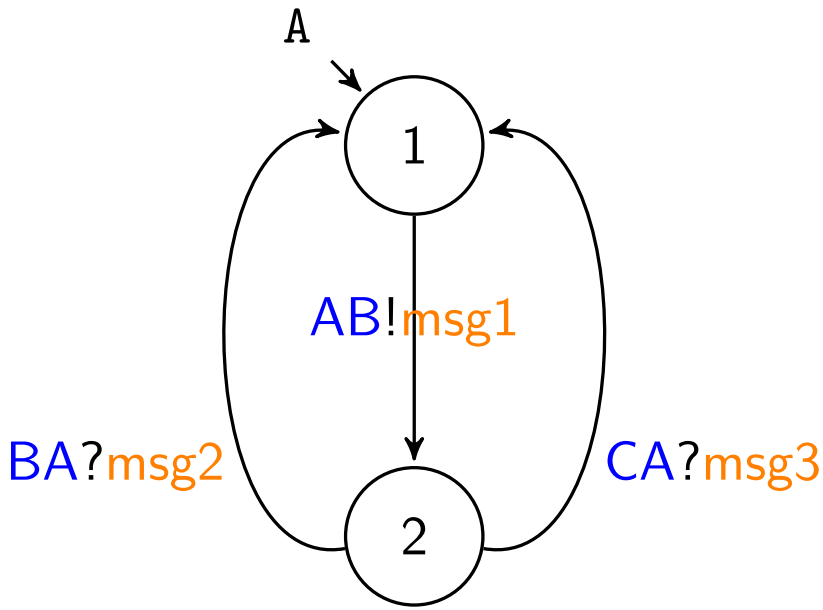


- $M_{\mathtt{A}}$ can send msg1 to machine $M_{\mathtt{B}}$;
  asynchronously; *through the directed buffered FIFO channel* AB
- Then, either msg2 or msg3 can be received from $M_{\mathtt{B}}$ or $M_{\mathtt{C}}$;
  through channels BA or CA;
- and so on....

Communicating Finite State Machines (CFSMs)

A formalism for the description and the analysis of distributed systems.

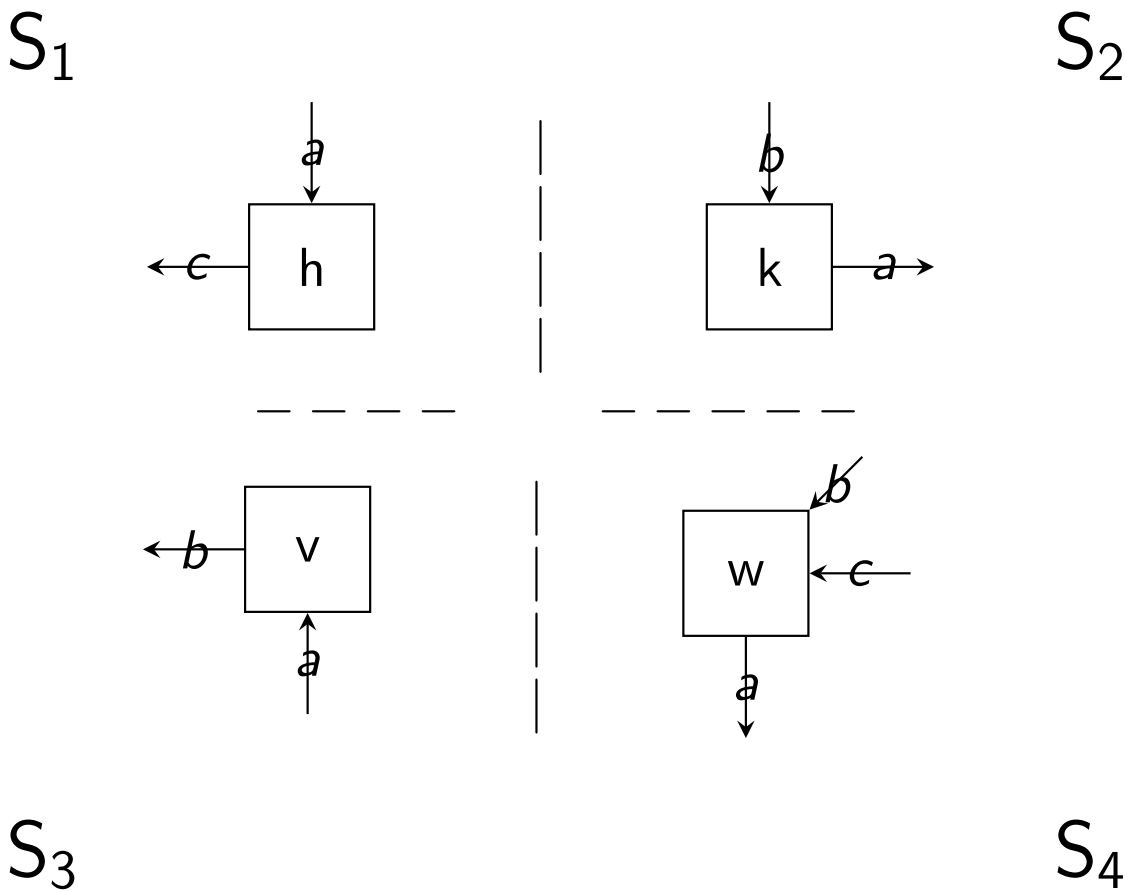
A machine $M_{\mathtt{A}}$



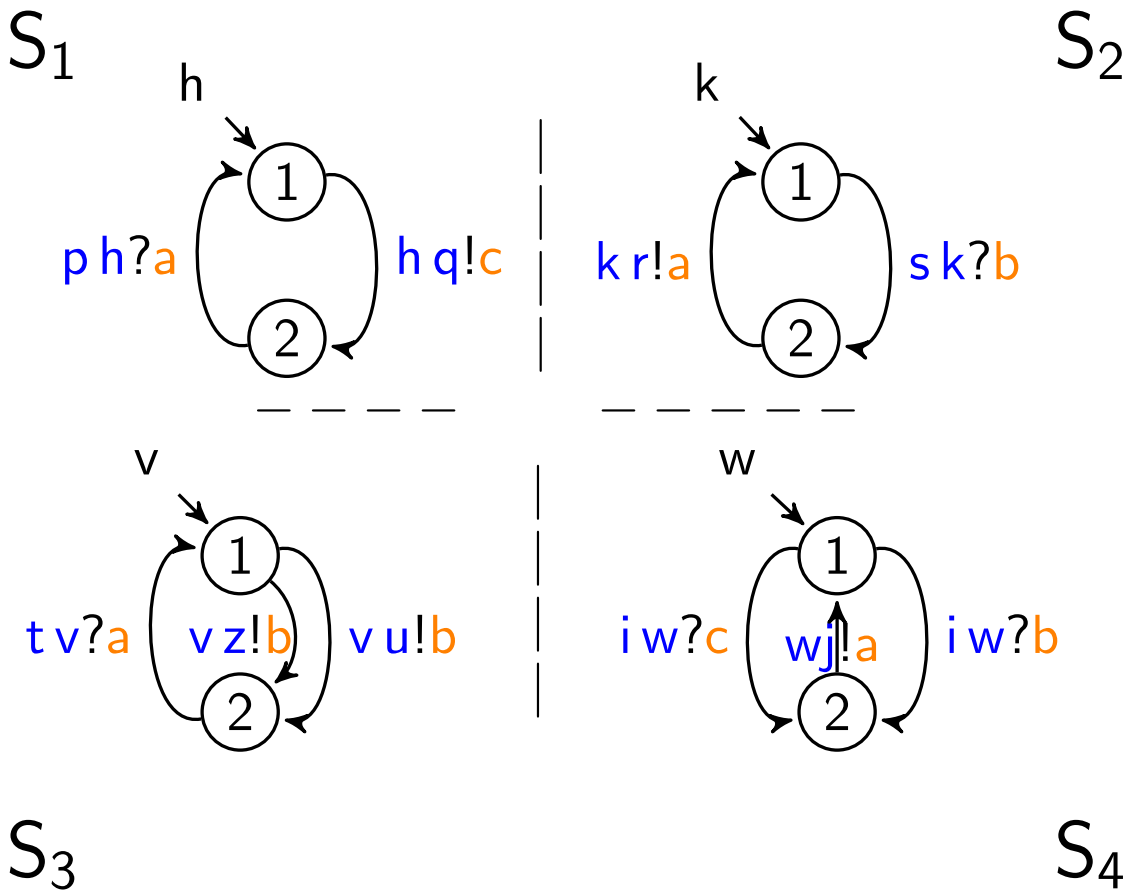
- $M_{\mathtt{A}}$ can send msg1 to machine $M_{\mathtt{B}}$;
  asynchronously; *through the directed buffered FIFO channel* AB
- Then, either msg2 or msg3 can be received from $M_{\mathtt{B}}$ or $M_{\mathtt{C}}$;
  through channels BA or CA;
- and so on....

The PaI approach for systems of CFSMs

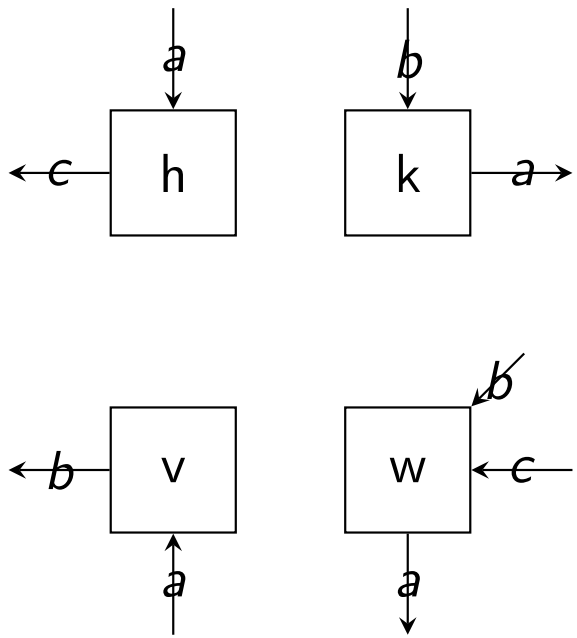
The PaI approach for systems of CFSMs



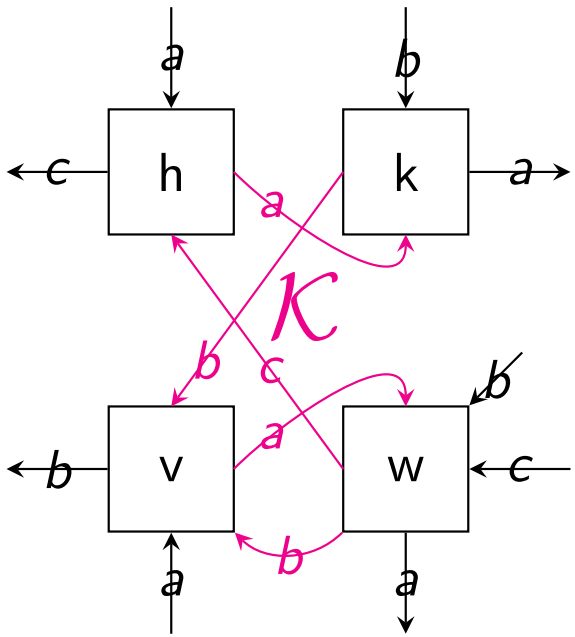
The PaI approach for systems of CFSMs



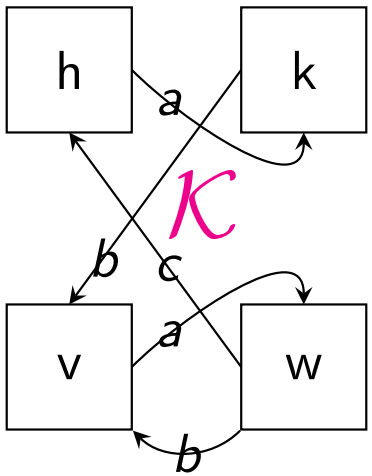
The PaI approach for systems of CFSMs



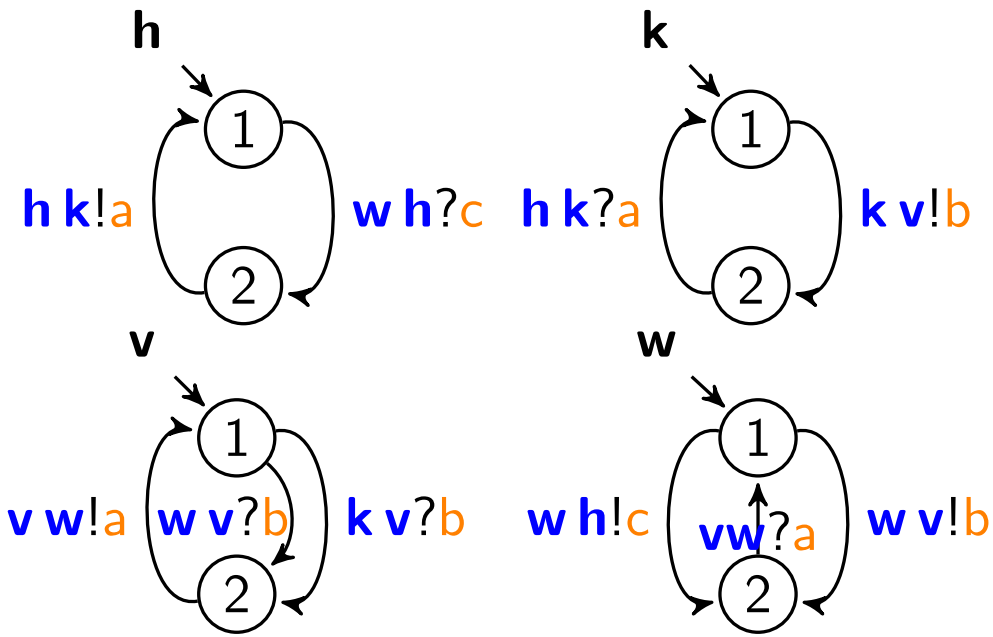
The PaI approach for systems of CFSMs



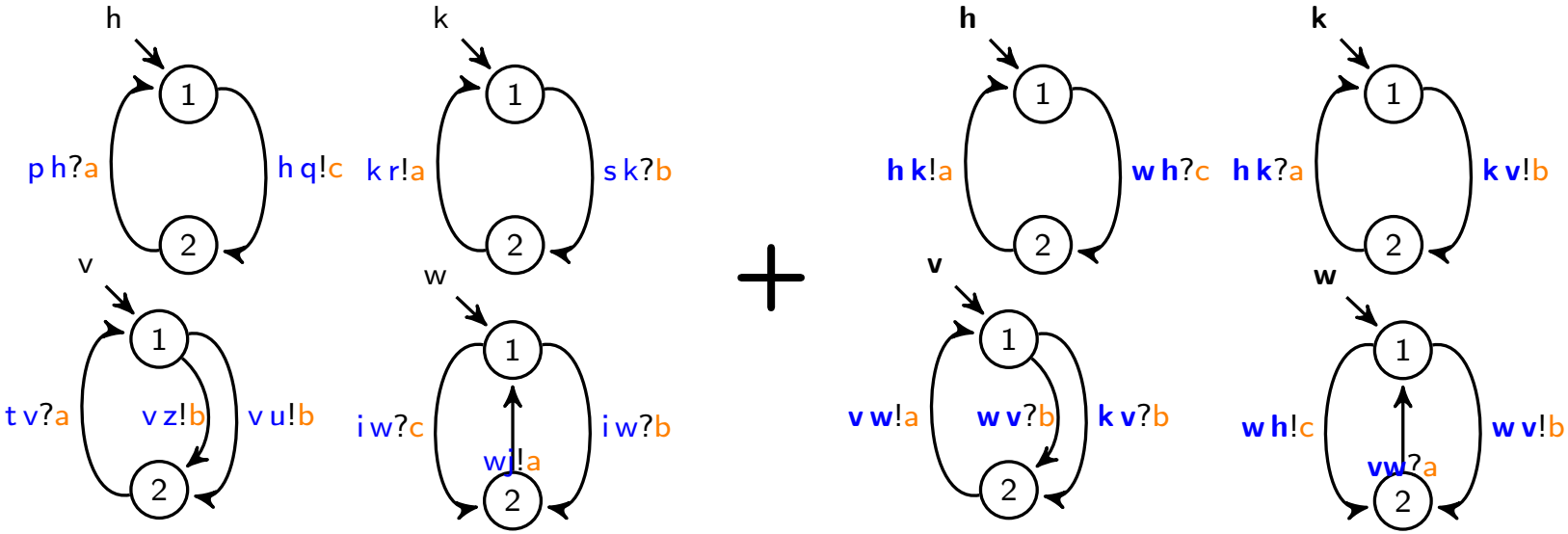
Connection Policies as CFSM systems



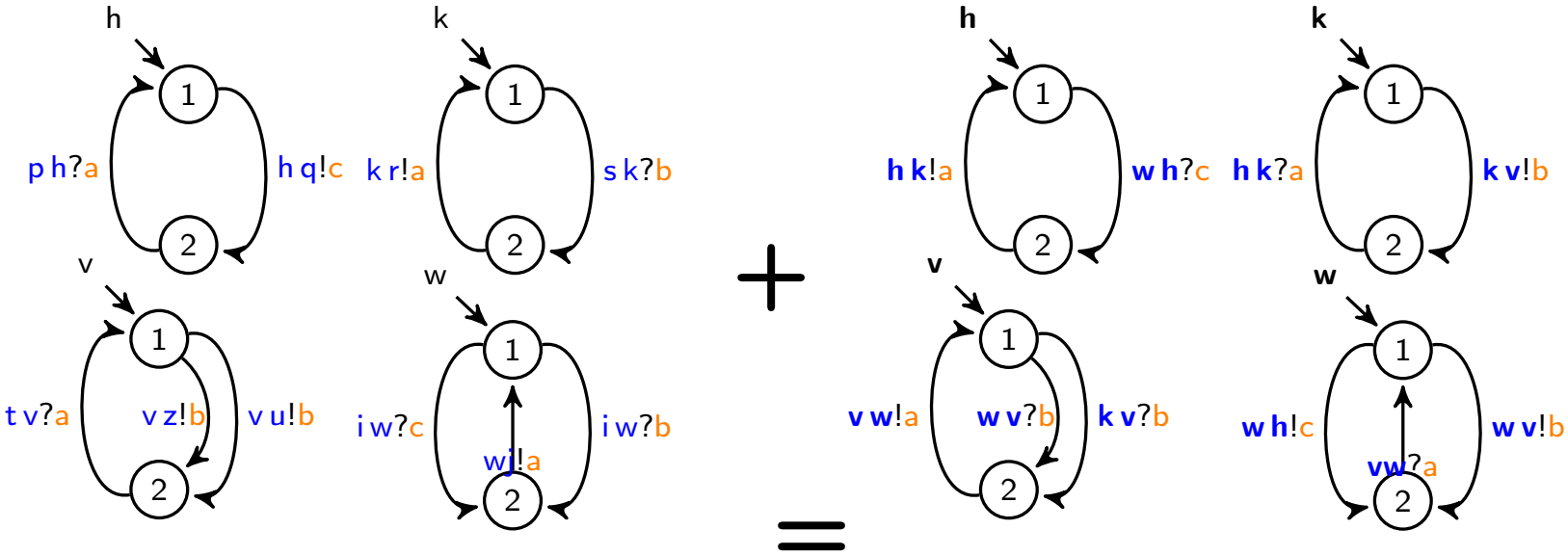
Connection policies are systems of CFSMs

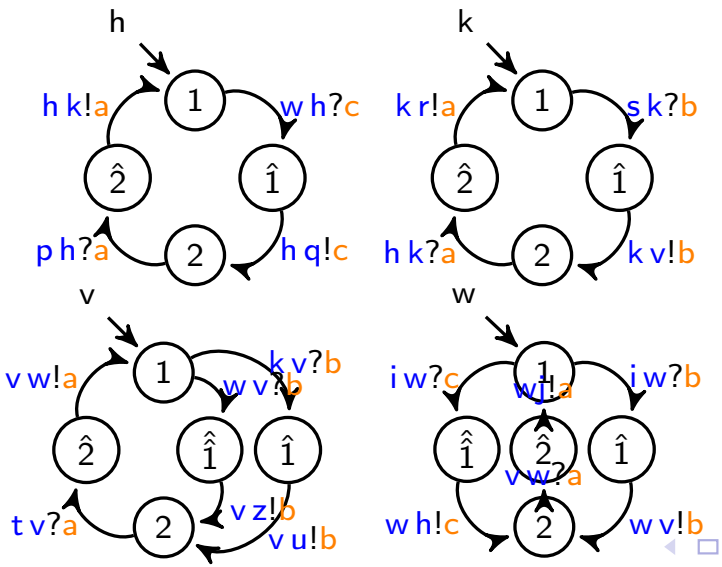


Interfaces + Connection Policy = Gateways



Interfaces + Connection Policy = Gateways





Safeness for PaI Multicomposition

Let
- $\{S_i\}_{i \in I}$ be a set of CFSM systems
- $\mathbb{K}$ a communication policy for choosen interfaces
- $\mathcal{P}$ be a communication property

Safeness for PaI Multicomposition

Let
- $\{S_i\}_{i \in I}$ be a set of CFSM systems
- $\mathbb{K}$ a communication policy for choosen interfaces
- $\mathcal{P}$ be a communication property

IF all S_i's and $\mathbb{K}$ enjoy $\mathcal{P}$ (and *no-mixed-state* holds)

Safeness for PaI Multicomposition

Let • $\{\mathsf{S}_i\}_{i\in I}$ be a set of CFSM systems
- $\mathbb{K}$ a communication policy for choosen interfaces
- $\mathcal{P}$ be a communication property

IF all S_i's and $\mathbb{K}$ enjoy $\mathcal{P}$ (and *no-mixed-state* holds)

THEN **mcomp**$(\{\mathsf{S}_i\}_{i\in I}, \mathbb{K})$ enjoys $\mathcal{P}$.

WHEN $\mathcal{P}$ is

Safeness for PaI Multicomposition

Let • $\{S_i\}_{i \in I}$ be a set of CFSM systems
- $\mathbb{K}$ a communication policy for choosen interfaces
- $\mathcal{P}$ be a communication property

IF all S_i's and $\mathbb{K}$ enjoy $\mathcal{P}$ (and *no-mixed-state* holds)

THEN **mcomp**$(\{S_i\}_{i \in I}, \mathbb{K})$ enjoys $\mathcal{P}$.

WHEN $\mathcal{P}$ is

- Deadlock-freeness ✓

Safeness for PaI Multicomposition

Let • $\{S_i\}_{i \in I}$ be a set of CFSM systems
- $\mathbb{K}$ a communication policy for choosen interfaces
- $\mathcal{P}$ be a communication property

IF all S_i's and $\mathbb{K}$ enjoy $\mathcal{P}$ (and *no-mixed-state* holds)

THEN **mcomp**$(\{S_i\}_{i \in I}, \mathbb{K})$ enjoys $\mathcal{P}$.

WHEN $\mathcal{P}$ is

- Deadlock-freeness ✓
- Orphan-message freeness ✓

Safeness for PaI Multicomposition

Let • $\{S_i\}_{i \in I}$ be a set of CFSM systems
- $\mathbb{K}$ a communication policy for choosen interfaces
- $\mathcal{P}$ be a communication property

IF all S_i's and $\mathbb{K}$ enjoy $\mathcal{P}$ (and *no-mixed-state* holds)

THEN **mcomp**$(\{S_i\}_{i \in I}, \mathbb{K})$ enjoys $\mathcal{P}$.

WHEN $\mathcal{P}$ is

- Deadlock-freeness ✓
- Orphan-message freeness ✓
- Reception-error freeness ✓

Safeness for PaI Multicomposition

Let • $\{S_i\}_{i \in I}$ be a set of CFSM systems
- $\mathbb{K}$ a communication policy for choosen interfaces
- $\mathcal{P}$ be a communication property

IF all S_i's and $\mathbb{K}$ enjoy $\mathcal{P}$ (and *no-mixed-state* holds)

THEN **mcomp**$(\{S_i\}_{i \in I}, \mathbb{K})$ enjoys $\mathcal{P}$.

WHEN $\mathcal{P}$ is

- Deadlock-freeness ✓
- Orphan-message freeness ✓
- Reception-error freeness ✓
- Progress ✓

Safeness for PaI Multicomposition

Let • $\{S_i\}_{i \in I}$ be a set of CFSM systems
- $\mathbb{K}$ a communication policy for choosen interfaces
- $\mathcal{P}$ be a communication property

IF all S_i's and $\mathbb{K}$ enjoy $\mathcal{P}$ (and *no-mixed-state* holds)

THEN **mcomp**$(\{S_i\}_{i \in I}, \mathbb{K})$ enjoys $\mathcal{P}$.

WHEN $\mathcal{P}$ is

- Deadlock-freeness ✓
- Orphan-message freeness ✓
- Reception-error freeness ✓
- Progress ✓
- Lock-freeness ✗

In the future

- Multicompatibility.
  In the binary composition:
  Compatibility of interfaces ($\sim$ bisimilarity) entails communication properties for the unique connection policy (if any).
- Multiconnection via "interfacing infrastructure".
- "Partial" gateways
  (some messages dealt with directly by the interfaces).
- How can Lock-freedom be recovered?

In the future

- Multicompatibility.
  In the binary composition:
  Compatibility of interfaces ($\sim$ bisimilarity) entails communication properties for the unique connection policy (if any).
- Multiconnection via "interfacing infrastructure".
- "Partial" gateways
  (some messages dealt with directly by the interfaces).
- How can Lock-freedom be recovered?

In the future

- Multicompatibility.
  In the binary composition:
  Compatibility of interfaces ($\sim$ bisimilarity) entails communication properties for the unique connection policy (if any).
- Multiconnection via "interfacing infrastructure".
- "Partial" gateways
  (some messages dealt with directly by the interfaces).
- How can Lock-freedom be recovered?

In the future

- Multicompatibility.
  In the binary composition:
  Compatibility of interfaces ($\sim$ bisimilarity) entails communication properties for the unique connection policy (if any).
- Multiconnection via "interfacing infrastructure".
- "Partial" gateways
  (some messages dealt with directly by the interfaces).
- How can Lock-freedom be recovered?

In the future

- Multicompatibility.
  In the binary composition:
  Compatibility of interfaces ($\sim$ bisimilarity) entails communication properties for the unique connection policy (if any).
- Multiconnection via "interfacing infrastructure".
- "Partial" gateways
  (some messages dealt with directly by the interfaces).
- How can Lock-freedom be recovered?

In the future

- Multicompatibility.
  In the binary composition:
  Compatibility of interfaces ($\sim$ bisimilarity) entails communication properties for the unique connection policy (if any).
- Multiconnection via "interfacing infrastructure".
- "Partial" gateways
  (some messages dealt with directly by the interfaces).
- How can Lock-freedom be recovered?

Thanks for your attention