Safe Composition of Systems of Communicating Finite State Machines

Franco Barbanera

University of Catania

APM@ETAPS'26 - April 16-17, 2026, Torino

OVERVIEW

- Introduction: the need for system composability
- The "participants-as-interfaces" (PaI) approach to **binary** system composition
- From binary PaI to: - multicomposition;
  - orchestrated multicomposition;
  - composition via partial gateways;
- Exploiting the PaI approach in the asynchronous framework of **C**ommunicating **F**inite **S**tate **M**achines.
- The essential of PaI: (partial) binary fusion.

Mainly based on works with ROLF HENNICKER and
on a paper with ROLF HENNICKER and U.DE'LIGUORO

OVERVIEW

- Introduction: the need for system composability
- The "participants-as-interfaces" (PaI) approach to **binary** system composition
- From binary PaI to: - multicomposition;
  - orchestrated multicomposition;
  - composition via partial gateways;
- Exploiting the PaI approach in the asynchronous framework of **C**ommunicating **F**inite **S**tate **M**achines.
- The essential of PaI: (partial) binary fusion.

Mainly based on works with ROLF HENNICKER and
on a paper with ROLF HENNICKER and U.DE'LIGUORO

OVERVIEW

- Introduction: the need for system composability
- The "participants-as-interfaces" (PaI) approach to **binary** system composition
- From binary PaI to: - multicomposition;
  - orchestrated multicomposition;
  - composition via partial gateways;
- Exploiting the PaI approach in the asynchronous framework of **C**ommunicating **F**inite **S**tate **M**achines.
- The essential of PaI: (partial) binary fusion.

Mainly based on works with ROLF HENNICKER and
on a paper with ROLF HENNICKER and U.DE'LIGUORO

OVERVIEW

- Introduction: the need for system composability
- The "participants-as-interfaces" (PaI) approach to **binary** system composition
- From binary PaI to: - multicomposition;
  - orchestrated multicomposition;
  - composition via partial gateways;
- Exploiting the PaI approach in the asynchronous framework of **C**ommunicating **F**inite **S**tate **M**achines.
- The essential of PaI: (partial) binary fusion.

Mainly based on works with ROLF HENNICKER and
on a paper with ROLF HENNICKER and U.DE'LIGUORO

OVERVIEW

- Introduction: the need for system composability
- The "participants-as-interfaces" (PaI) approach to **binary** system composition
- From binary PaI to: - multicomposition;
  - orchestrated multicomposition;
  - composition via partial gateways;
- Exploiting the PaI approach in the asynchronous framework of **C**ommunicating **F**inite **S**tate **M**achines.
- The essential of PaI: (partial) binary fusion.

Mainly based on works with ROLF HENNICKER and
on a paper with ROLF HENNICKER and U.DE'LIGUORO

OVERVIEW

- Introduction: the need for system composability
- The "participants-as-interfaces" (PaI) approach to **binary** system composition
- From binary PaI to: - multicomposition;
  - orchestrated multicomposition;
  - composition via partial gateways;
- Exploiting the PaI approach in the asynchronous framework of **C**ommunicating **F**inite **S**tate **M**achines.
- The essential of PaI: (partial) binary fusion.

Mainly based on works with ROLF HENNICKER and
on a paper with ROLF HENNICKER and U.DE'LIGUORO

OVERVIEW

- Introduction: the need for system composability
- The "participants-as-interfaces" (PaI) approach to **binary** system composition
- From binary PaI to: - multicomposition;
  - orchestrated multicomposition;
  - composition via partial gateways;
- Exploiting the PaI approach in the asynchronous framework of **C**ommunicating **F**inite **S**tate **M**achines.
- The essential of PaI: (partial) binary fusion.

Mainly based on works with Rolf Hennicker and
on a paper with Rolf Hennicker and U.de'Liguoro

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,
Connecting open systems of communicating finite state machines.
J. Log. Algebraic Methods in Program. (2019)

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

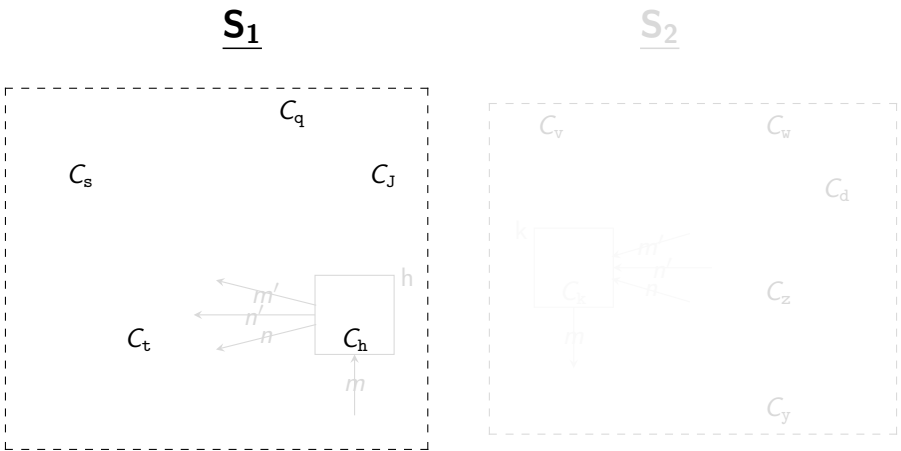
applicable when

participant behaviour  **system interface**

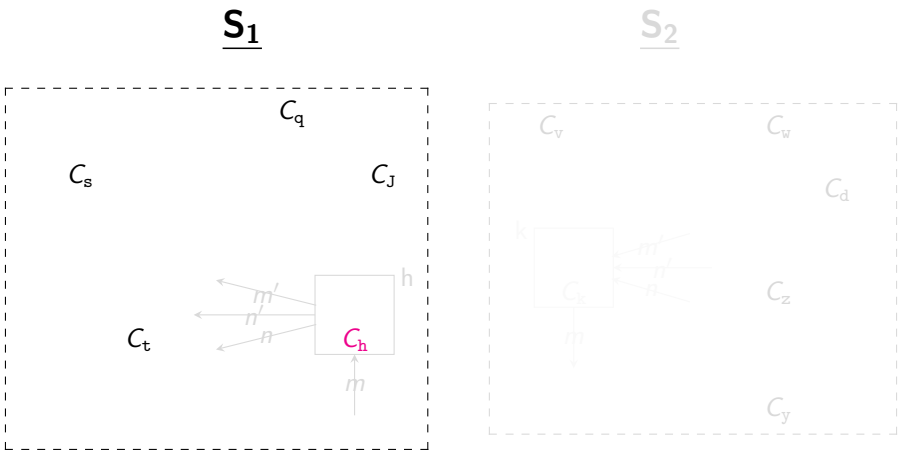
where

interface = description of possible interactions with an outer system.

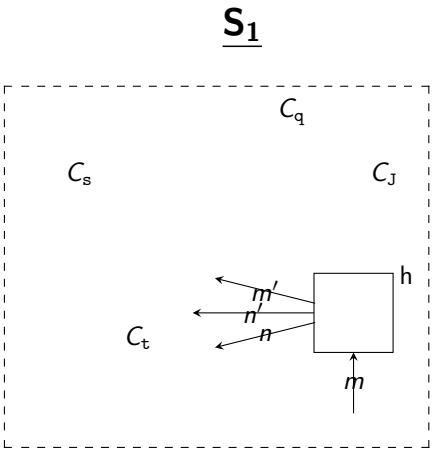
The "participants-as-interfaces" (PaI) approach



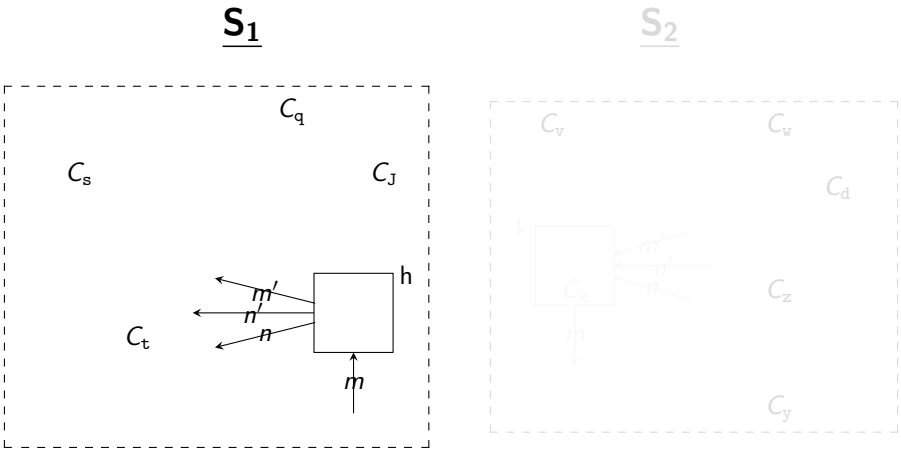
The "participants-as-interfaces" (PaI) approach



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

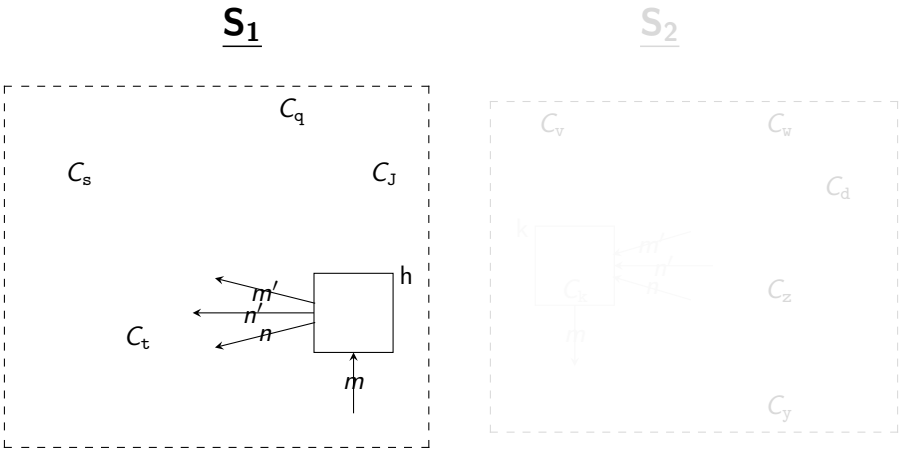


The "participants-as-interfaces" (PaI) approach



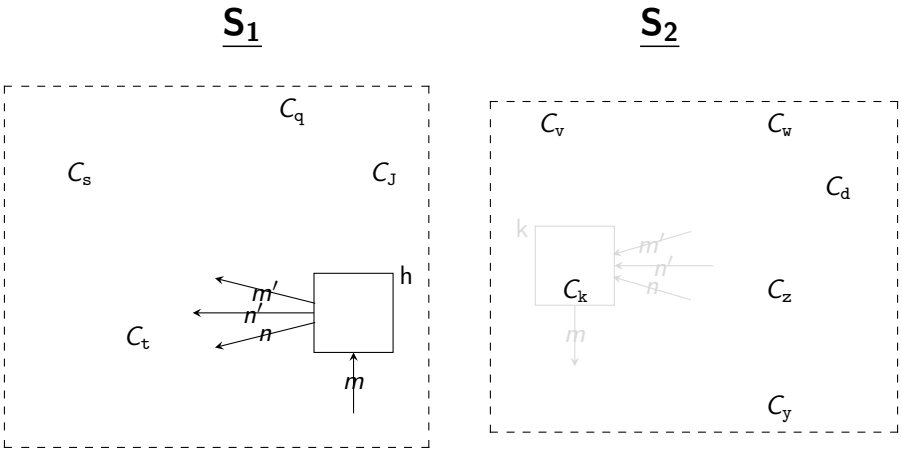
We abstract here from the way communications are performed and from the logical order of the exchanged messages.

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

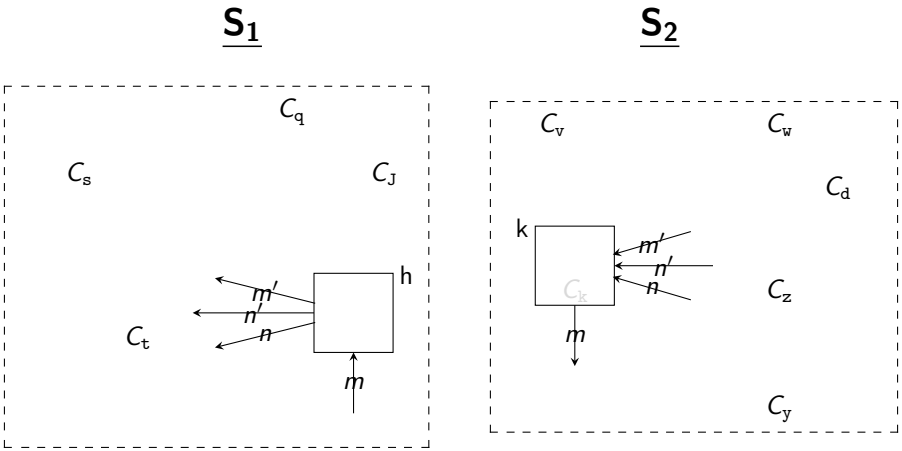


C_h's behaviour can be looked at as an interface (i.e. a description of what can be offered by an outer system)

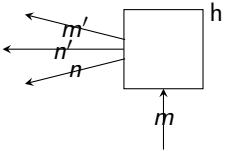
The "participants-as-interfaces" (PaI) approach

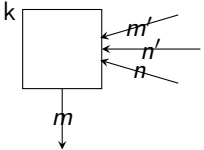


The "participants-as-interfaces" (PaI) approach



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

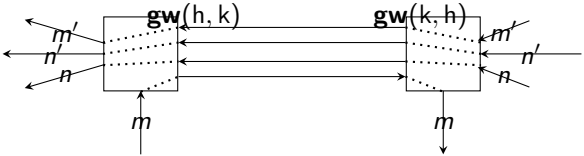




The "participants-as-interfaces" (PaI) approach

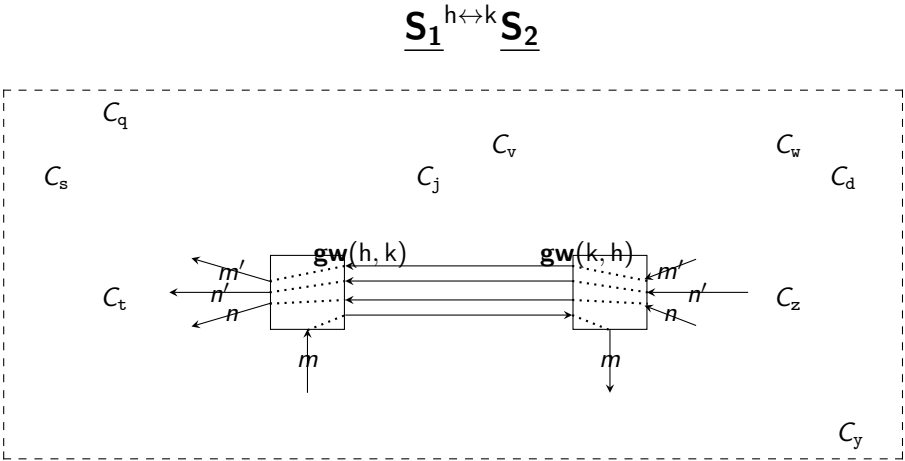
COMPATIBLE: an h's input is a k's output, and vice versa

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



Composition via gateways (forwarders)

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



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

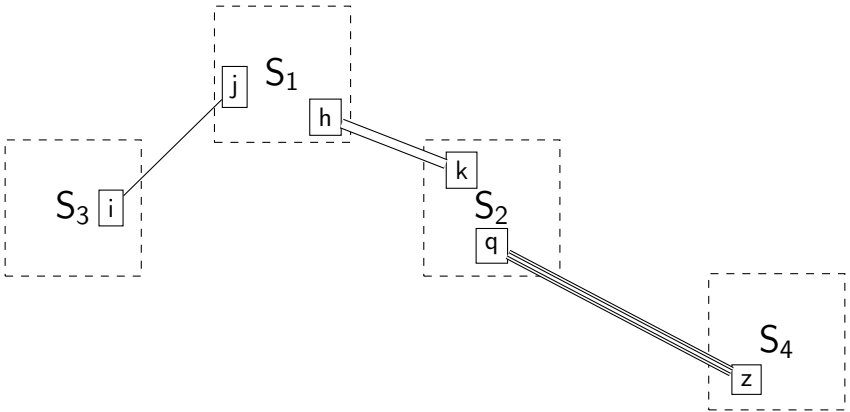
Drawback of binary composition:

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.

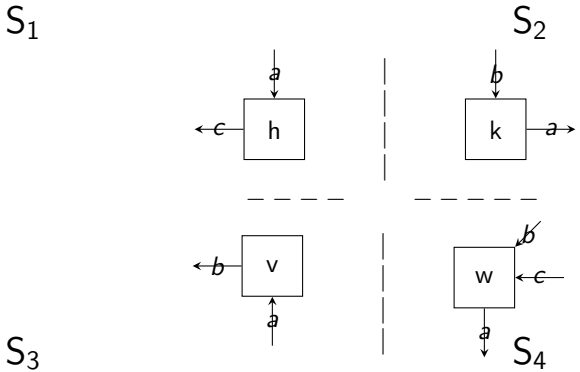


PaI Multicomposition

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

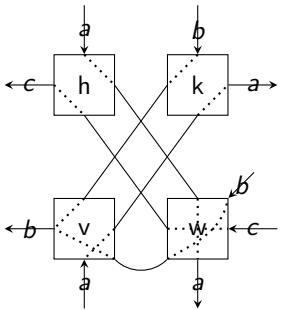
PaI Multicomposition

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



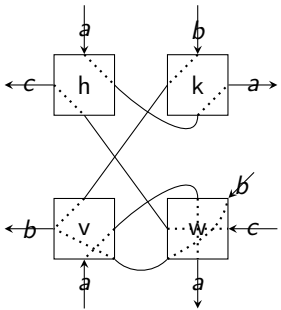
PaI Multicomposition

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



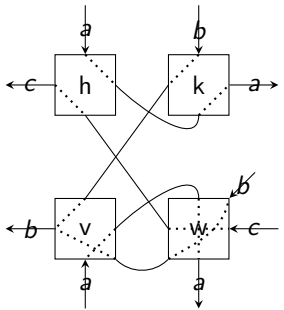
PaI Multicomposition

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



PaI Multicomposition

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

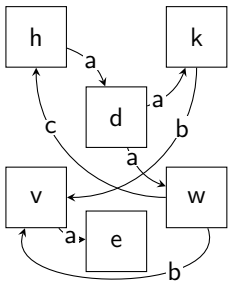


Issues: • Many different "connection policies" (all safe?).

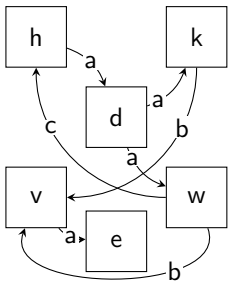
Connection policies as systems

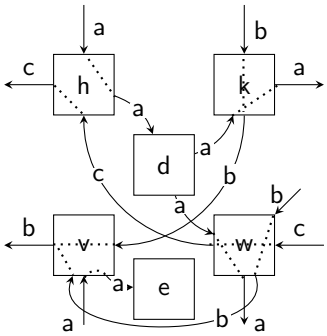
Gateways are built out of the interface participants and a choosen connection policy

Orchestrated connection policies

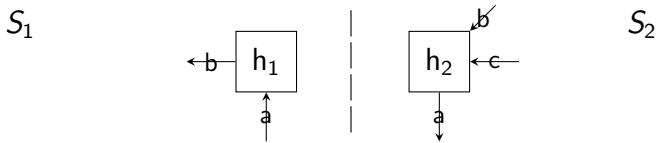


Orchestrated connection policies

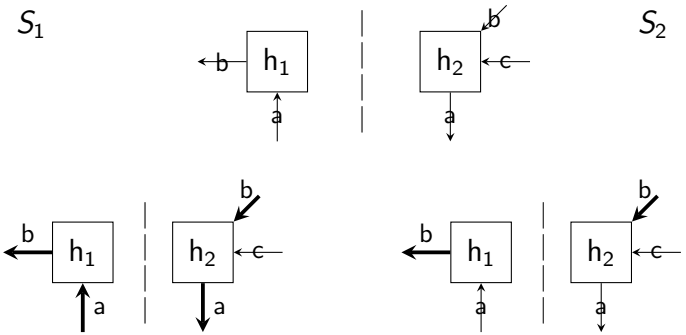




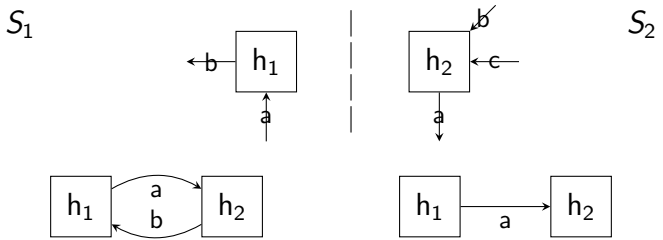
PaI Composition via **partial** gateways



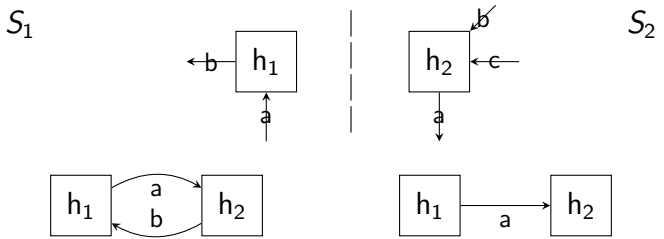
PaI Composition via **partial** gateways



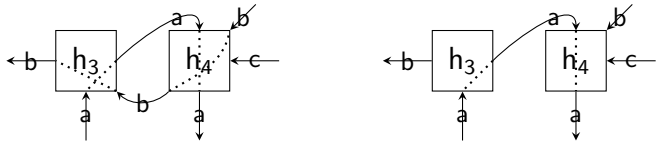
PaI Composition via **partial** gateways



Pal Composition via **partial** gateways



The above connection policies lead to the following partial gateways



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

It depends!

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

- SAFE

It depends!

- On the concurrent-system description formalism

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

- SAFE

It depends!

- On the concurrent-system description formalism
- On the communication model

Safety in Synchronous formalisms

Rather severe restrictions in both

- **M**ulti**P**arty **S**ession **T**ypes
  - F. Barbanera, M. Dezani-Ciancaglini, I. Lanese, E. Tuosto: Composition and decomposition of multiparty sessions. JLAMP (2021)
  - F. Barbanera, M. Dezani-Ciancaglini, L. Gheri, N. Yoshida: Multicompatibility for Multiparty-Session Composition. PPDP 2023
- **synchronous** version of **CFSM**
  - F. Barbanera, Ivan Lanese, Emilio Tuosto: Composition of synchronous communicating systems. JLAMP (2023)

Exploiting the PaI approach

in

(**asynchronous**)
Communicating **F**inite **S**tate **M**achines

Communicating Finite State Machines (CFSMs)

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

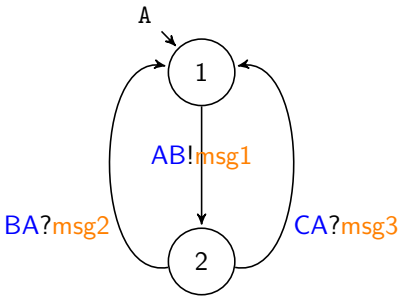
A machine M_A

- 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_A

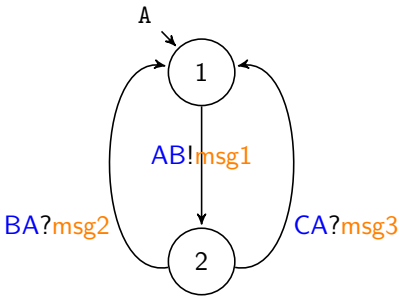


- M_A can send msg1 to machine M_B;
  asynchronously; *through the directed buffered FIFO channel* AB
- Then, either msg2 or msg3 can be received from M_B or M_C;
  through channels BA or CA;
- and so on....

Communicating Finite State Machines (CFSMs)

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

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

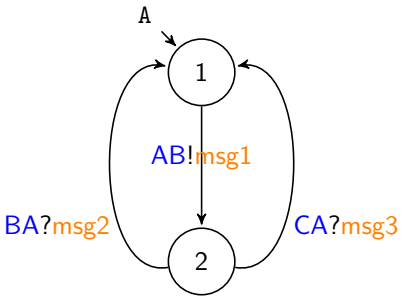


- $M_{\mathtt{A}}$ can send msg1 to machine $M_{\mathtt{B}}$;
  asynchronously; *through the directed buffered FIFO channel* AB
- Then, either msg2 or msg3 can be received from $M_{\mathtt{B}}$ or $M_{\mathtt{C}}$;
  through channels BA or CA;
- and so on....

Communicating Finite State Machines (CFSMs)

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

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

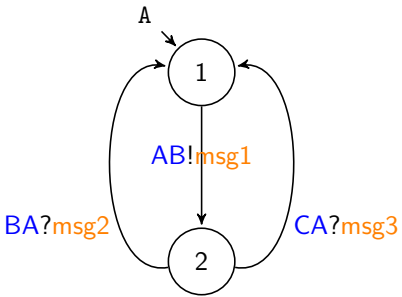


- $M_{\mathtt{A}}$ can send msg1 to machine $M_{\mathtt{B}}$;
  asynchronously; *through the directed buffered FIFO channel* AB
- Then, either msg2 or msg3 can be received from $M_{\mathtt{B}}$ or $M_{\mathtt{C}}$;
  through channels BA or CA;
- and so on....

Communicating Finite State Machines (CFSMs)

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

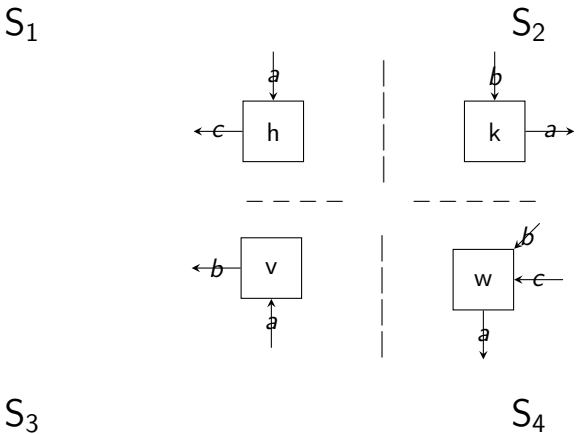
A machine $M_{\mathtt{A}}$



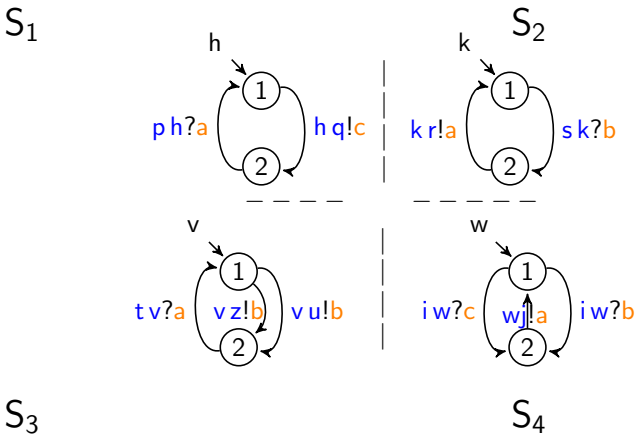
- $M_{\mathtt{A}}$ can send msg1 to machine $M_{\mathtt{B}}$;
  asynchronously; *through the directed buffered FIFO channel* AB
- Then, either msg2 or msg3 can be received from $M_{\mathtt{B}}$ or $M_{\mathtt{C}}$;
  through channels BA or CA;
- and so on....

The PaI approach for systems of CFSMs

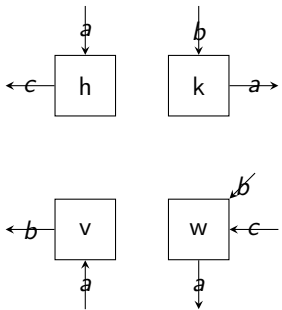
The PaI approach for systems of CFSMs



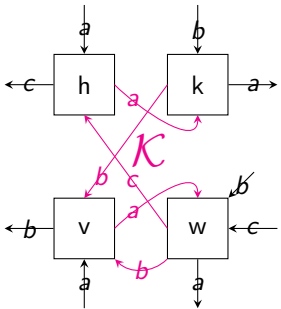
The PaI approach for systems of CFSMs



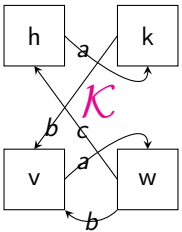
The PaI approach for systems of CFSMs



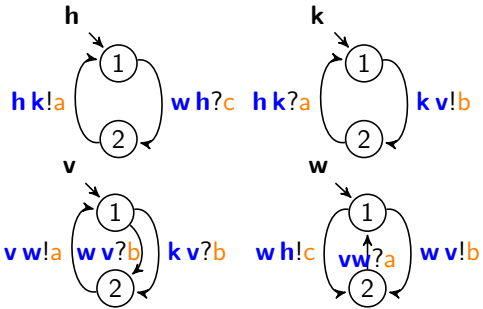
The PaI approach for systems of CFSMs



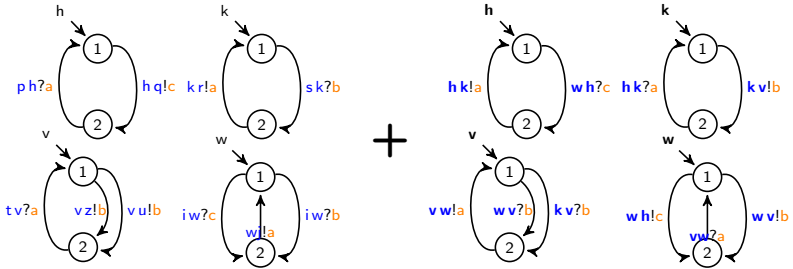
Connection Policies as CFSM systems



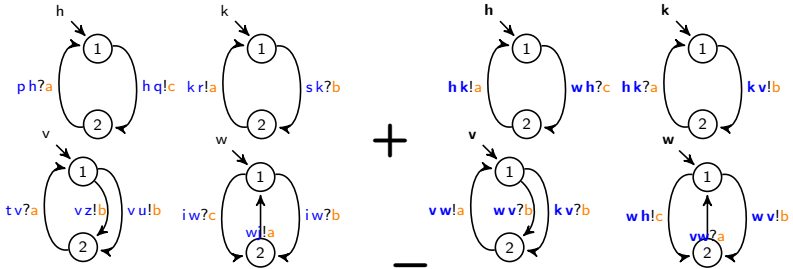
Connection policies are systems of CFSMs



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 • $\{S_i\}_{i \in I}$ 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

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-freedom ✓

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-freedom ✓
- Orphan-message freedom ✓

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-freedom ✓
- Orphan-message freedom ✓
- Reception-error freedom ✓

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-freedom ✓
- Orphan-message freedom ✓
- Reception-error freedom ✓
- 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-freedom ✓
- Orphan-message freedom ✓
- Reception-error freedom ✓
- Progress ✓
- Lock-freedom ✗

Safeness for PaI Orchestrated Multicomposition

Let • $\{S_i\}_{i \in I}$ be a set of CFSM systems
- $\mathbb{K}$ an orchestrated comm. 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 **OrchMcomp**$(\{S_i\}_{i \in I}, \mathbb{K})$ enjoys $\mathcal{P}$.

WHEN $\mathcal{P}$ is

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

Safeness for PaI composition via partial gateways

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

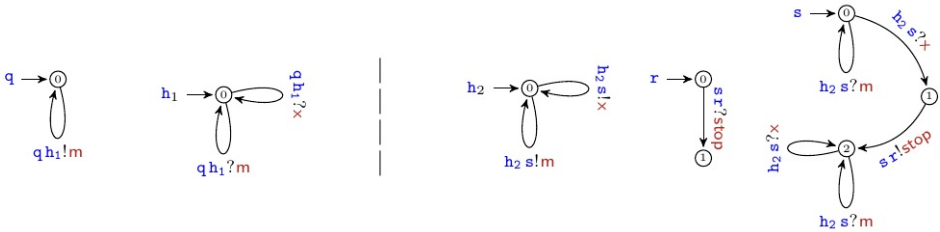
IF all S_i's and $\mathbb{K}$ enjoy $\mathcal{P}$ (and *no-mixed-state* holds)

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

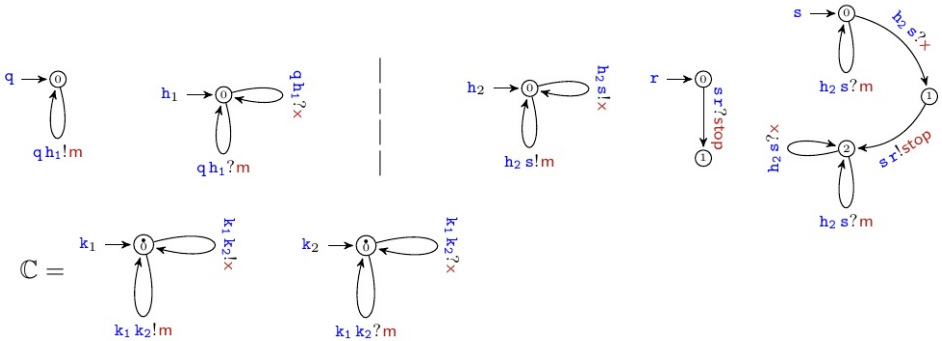
WHEN $\mathcal{P}$ is

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

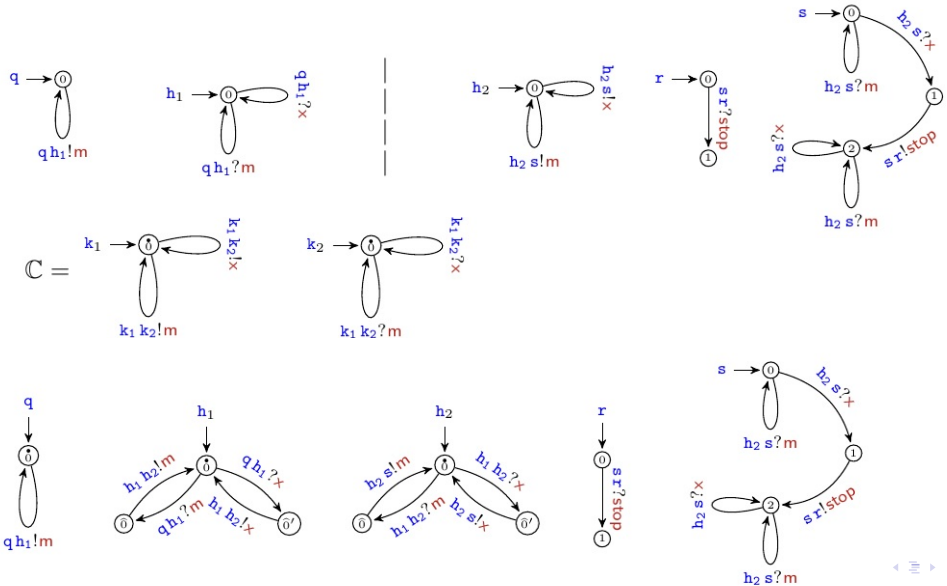
Lock-freedom counterexample



Lock-freedom counterexample



Lock-freedom counterexample



All proofs by contradiction

Based on
the notion of **projection**: configurations of original systems and of connection policy obtainable from a configuration of the composed system;

and on the fact that projection preserves (roughly) reachability

All preserved properties are safety properties.
So - roughly - problematic reachable configurations of original systems and of connection policy obtaineble from a problematic reachable configuration of the composed system

All proofs by contradiction

Based on
the notion of **projection**: configurations of original systems and of connection policy obtainable from a configuration of the composed system;

and on the fact that projection preserves (roughly) reachability

All preserved properties are safety properties.
So - roughly - problematic reachable configurations of original systems and of connection policy obtaineble from a problematic reachable configuration of the composed system

All proofs by contradiction

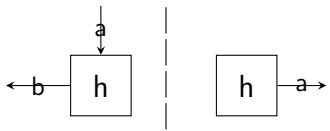
Based on
the notion of **projection**: configurations of original systems and of connection policy obtainable from a configuration of the composed system;

and on the fact that projection preserves (roughly) reachability

All preserved properties are safety properties.
So - roughly - problematic reachable configurations of original systems and of connection policy obtaineble from a problematic reachable configuration of the composed system

The basis of all:
Partial Fusion Composition (under peer-review)

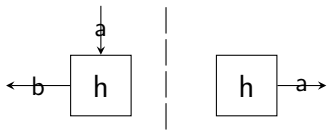
S_1



S_2

The basis of all:
Partial Fusion Composition (under peer-review)

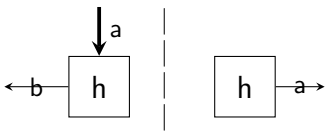
S_1



S_2

If some transitions of h in S_1 perfectly complement the transitions of h in S_2...

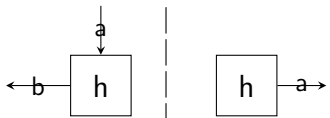
S_1



S_2

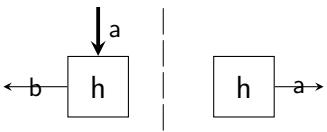
The basis of all:
Partial Fusion Composition (under peer-review)

S_1 S_2



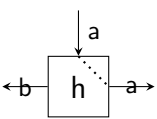
If some transitions of h in S_1 perfectly complement the transitions of h in S_2...

S_1 S_2



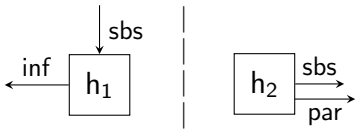
... then FUSE the two h's

$S_1 \overset{\text{h}}{\texttt{fuse}} S_2$



An example of binary PaI via partial gateways

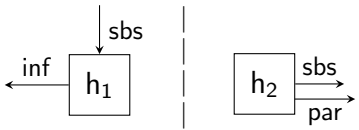
S_1



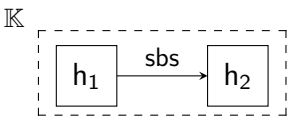
S_2

An example of binary PaI via partial gateways

S_1

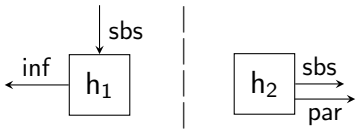


S_2



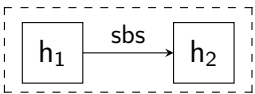
An example of binary PaI via partial gateways

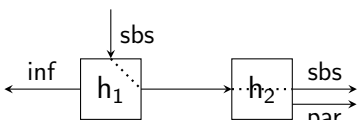
S_1



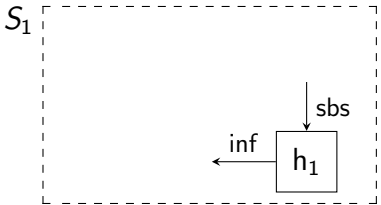
S_2

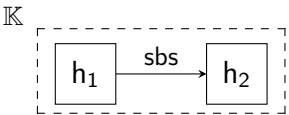
$\mathbb{K}$



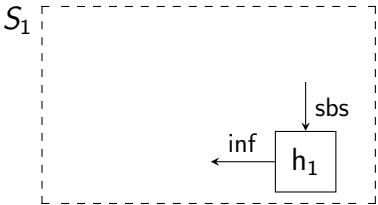


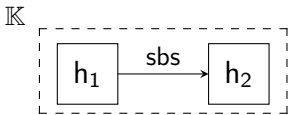
Binary PaI via partial gateways from Partial Fusion

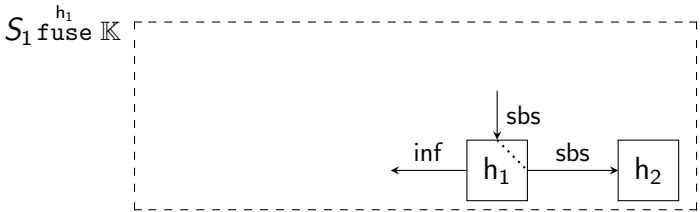




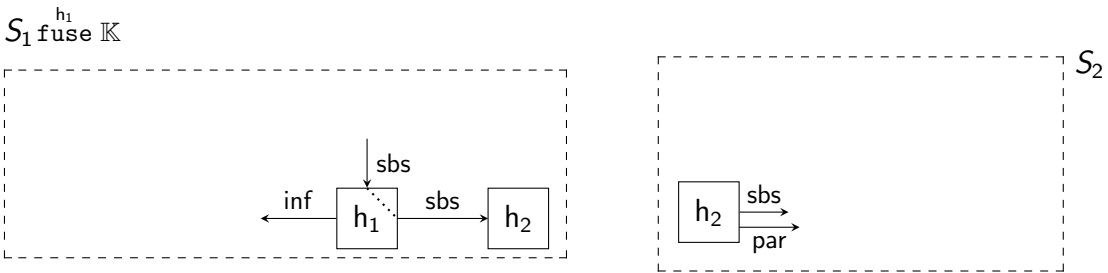
Binary PaI via partial gateways from Partial Fusion





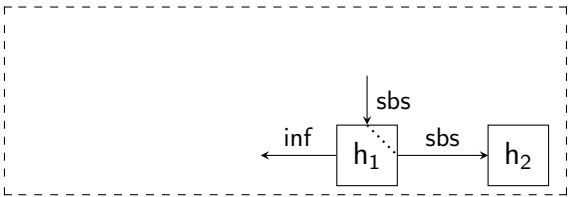


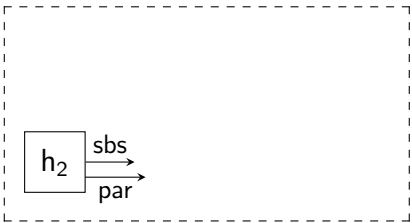
Binary PaI via partial gateways from Partial Fusion



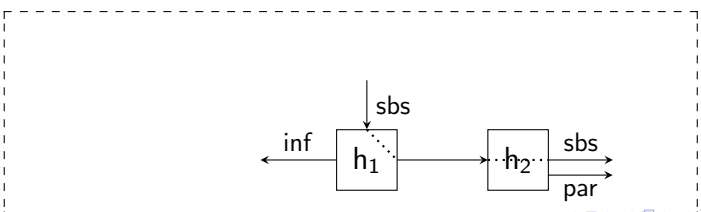
Binary PaI via partial gateways from Partial Fusion

$S_1 \overset{h_1}{\texttt{fuse}} \mathbb{K}$





$(S_1 \overset{h_1}{\texttt{fuse}}) \overset{h_2}{\texttt{fuse}} S_2$



In the future

- (easy) formally show that all the PaI multicompositions are obtainable from fusion composition:
- some (liveness?) conditions to get preservation of Lock-freedom for partial fusion composition:
- investigating PaI in frameworks other than CFSM.

In the future

- (easy) formally show that all the PaI multicompositions are obtainable from fusion composition:
- some (liveness?) conditions to get preservation of Lock-freedom for partial fusion composition:
- investigating PaI in frameworks other than CFSM.

In the future

- (easy) formally show that all the PaI multicompositions are obtainable from fusion composition:
- some (liveness?) conditions to get preservation of Lock-freedom for partial fusion composition:
- investigating PaI in frameworks other than CFSM.

In the future

- (easy) formally show that all the PaI multicompositions are obtainable from fusion composition:
- some (liveness?) conditions to get preservation of Lock-freedom for partial fusion composition:
- investigating PaI in frameworks other than CFSM.

Thanks for your attention