Asynchronous Message Orderings Beyond Causality

Adam Shimi, Aurélie Hurault, Philippe Quéinnec

IRIT, University of Toulouse

December 20, 2017

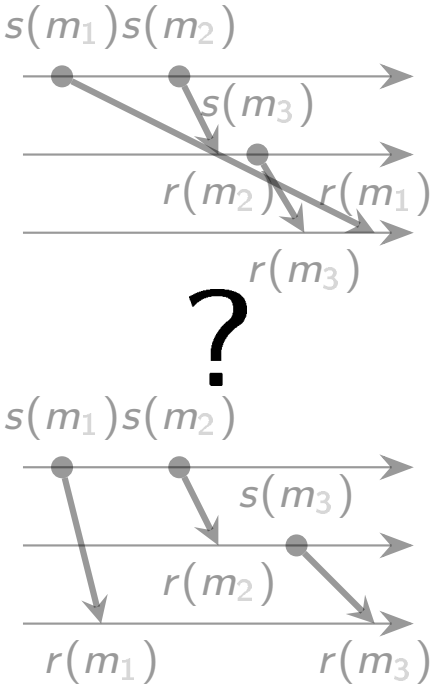
Université
de Toulouse

Table Of Contents

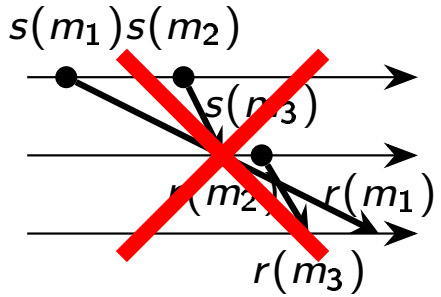
Table Of Contents

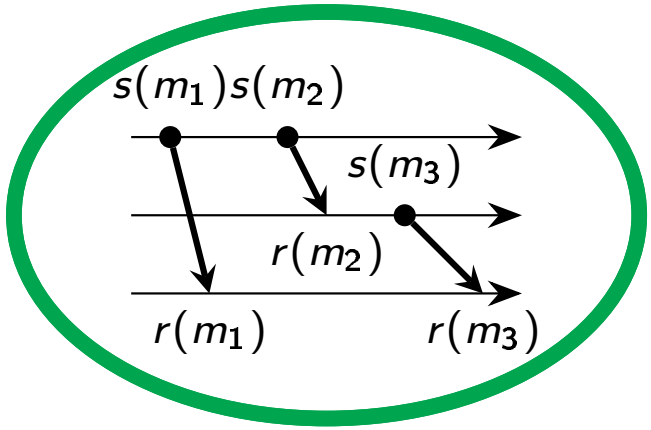
The Two Faces of Ordering

Detecting

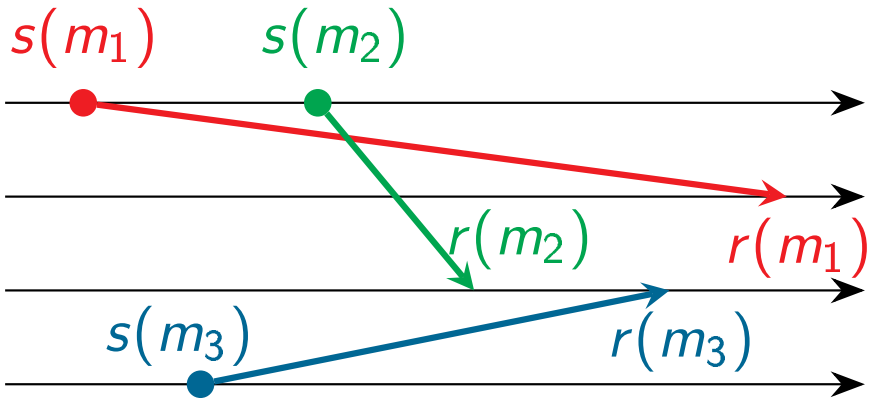


Enforcing

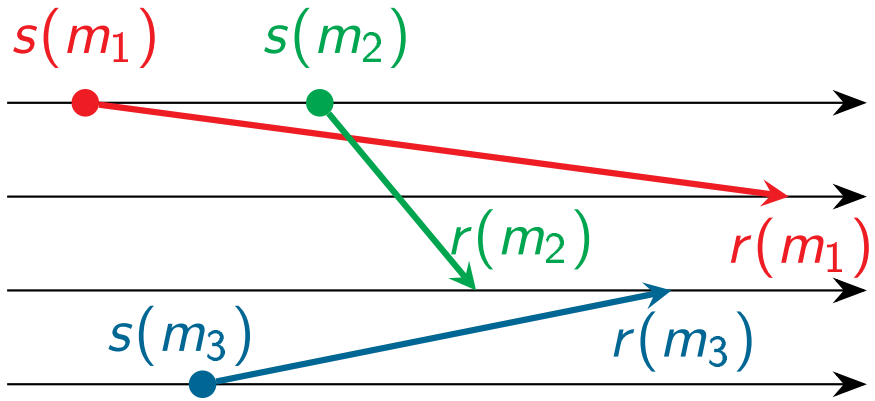


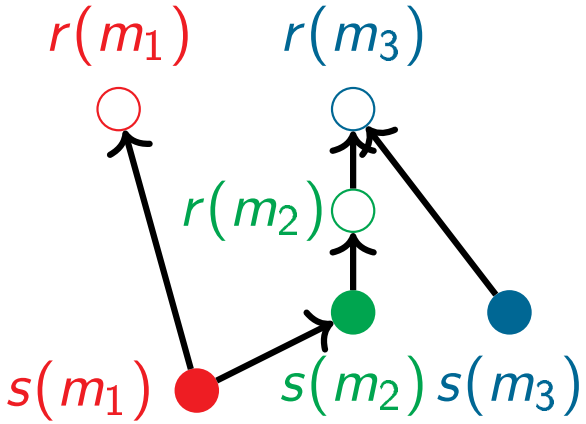


Computations and Executions



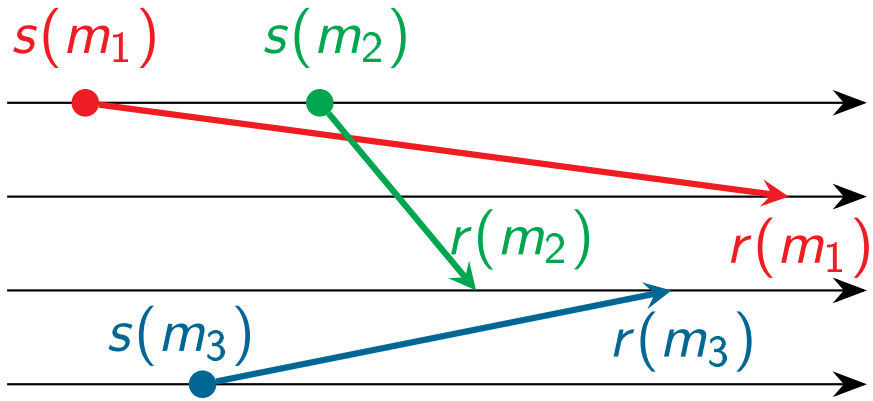
Computations and Executions

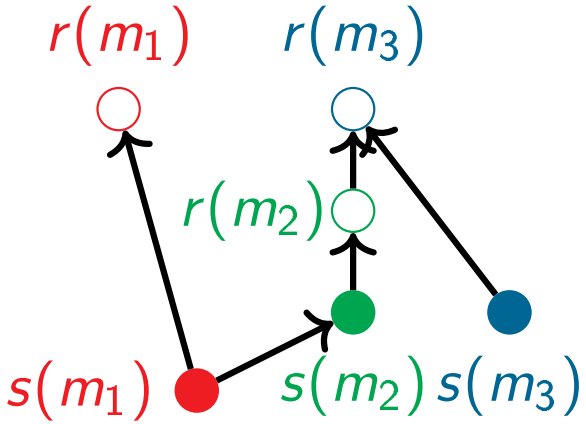




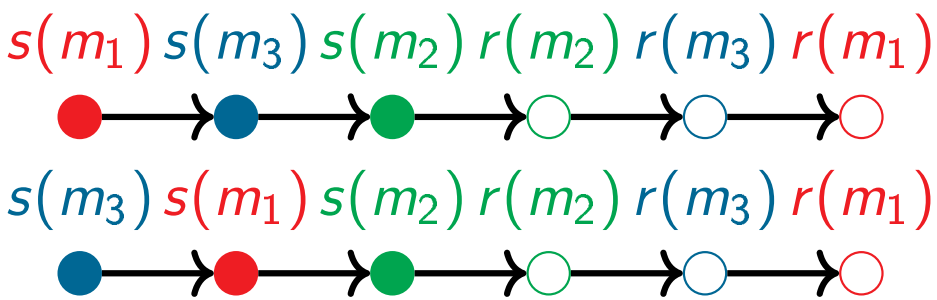
Computation $(\prec_c)$

Computations and Executions



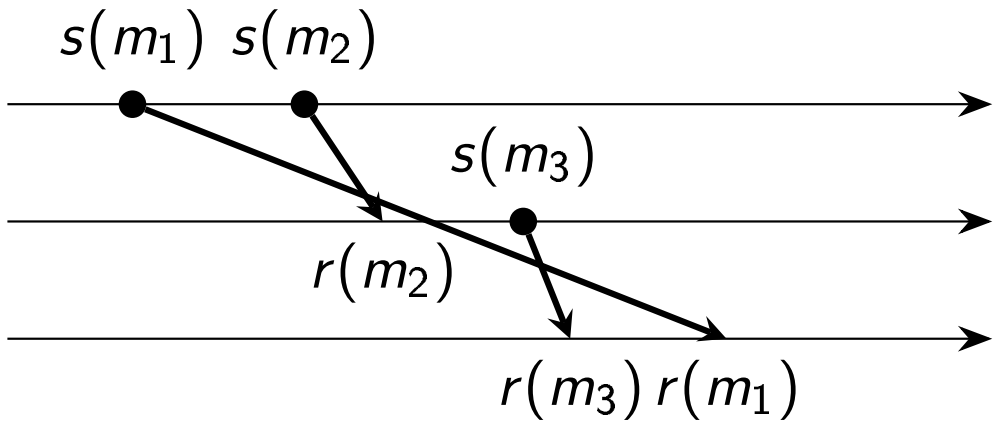


Computation $(\prec_c)$



Executions $(\prec)$

A Predicate on Computations: Causal Delivery

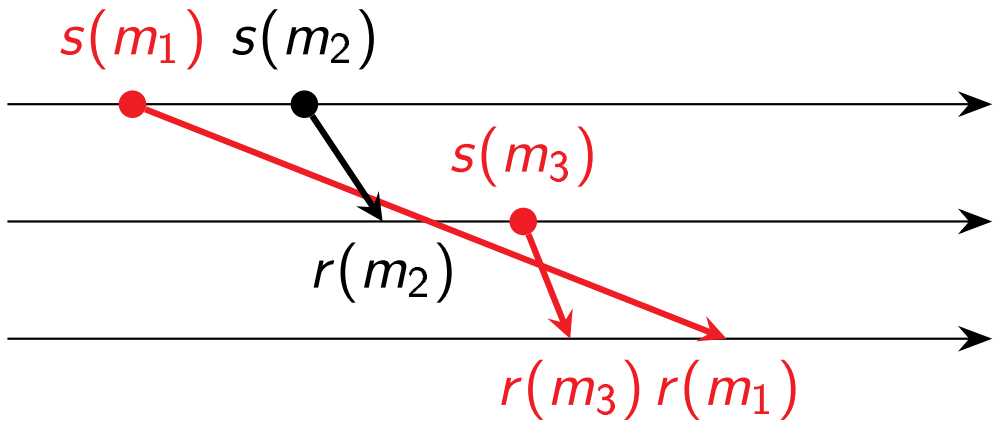


Causal Delivery

A computation x satisfies **Causal Delivery** $\triangleq$ for every m_1, m_2 sent to the same peer and received in x, we have

$$s(m_1) \prec_c s(m_2) \implies r(m_1) \prec_c r(m_2)$$

A Predicate on Computations: Causal Delivery

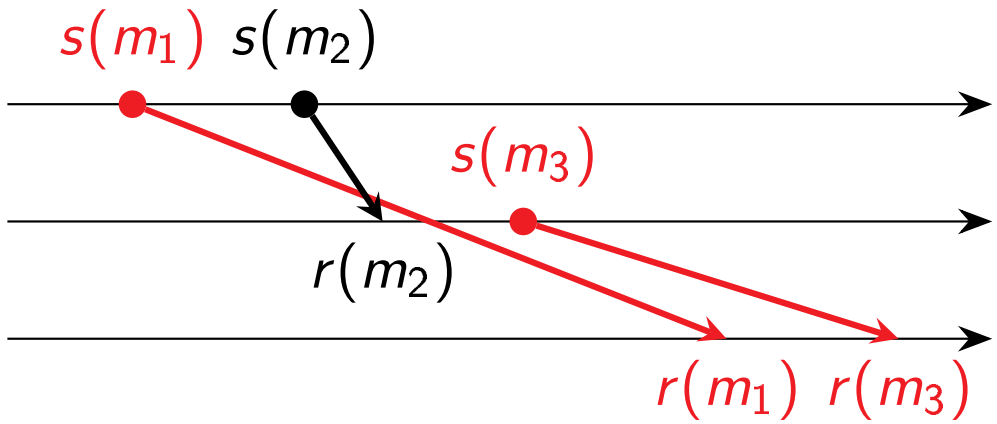


Causal Delivery

A computation x satisfies **Causal Delivery** $\triangleq$ for every m_1, m_2 sent to the same peer and received in x, we have

$$s(m_1) \prec_c s(m_2) \implies r(m_1) \prec_c r(m_2)$$

A Predicate on Computations: Causal Delivery

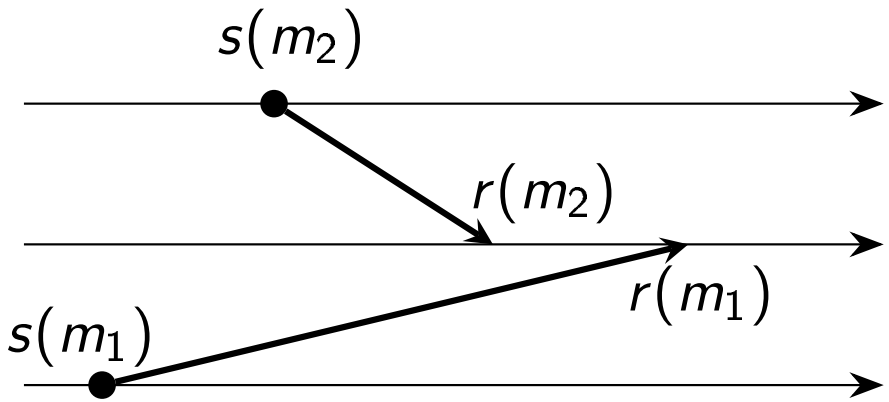


Causal Delivery

A computation x satisfies **Causal Delivery** $\triangleq$ for every m_1, m_2 sent to the same peer and received in x, we have

$$s(m_1) \prec_c s(m_2) \implies r(m_1) \prec_c r(m_2)$$

A Predicate on Executions: Fifo_{n-1} Delivery

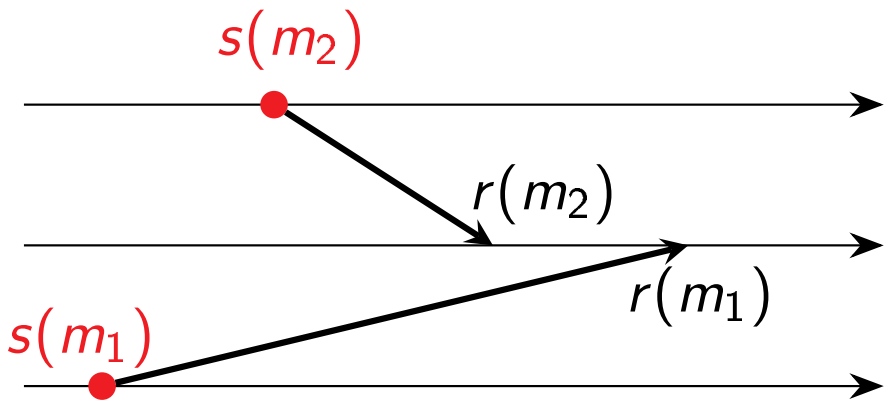


Fifo_{n-1} Delivery

An execution σ satisfies **Fifo_{n-1} Delivery** $\triangleq$ for every m_1, m_2 sent to the same peer and received in σ, we have

$$s(m_1) \prec s(m_2) \implies r(m_1) \prec_c r(m_2)$$

A Predicate on Executions: Fifo_{n-1} Delivery

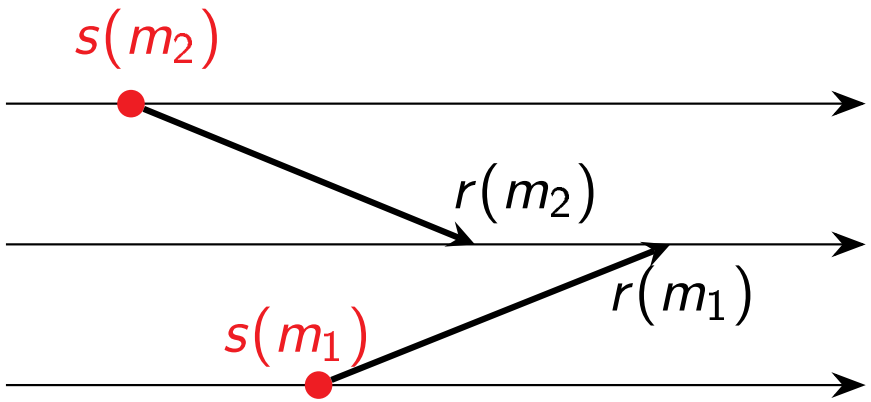


Fifo_{n-1} Delivery

An execution σ satisfies **Fifo_{n-1} Delivery** $\triangleq$ for every m_1, m_2 sent to the same peer and received in σ, we have

$$s(m_1) \prec s(m_2) \implies r(m_1) \prec_c r(m_2)$$

A Predicate on Executions: Fifo_{n-1} Delivery



Fifo_{n-1} Delivery

An execution σ satisfies **Fifo_{n-1} Delivery** $\triangleq$ for every m_1, m_2 sent to the same peer and received in σ, we have

$$s(m_1) \prec s(m_2) \implies r(m_1) \prec_c r(m_2)$$

Issue with Predicates on Executions

The problem

Asynchrony $\Longrightarrow$ distributed behavior = computation.

Yet some predicates on executions are not well-defined on computations (for example Fifo_{n-1} delivery).

Issue with Predicates on Executions

The problem

Asynchrony $\Longrightarrow$ distributed behavior = computation.

Yet some predicates on executions are not well-defined on computations (for example Fifo_{n-1} delivery).

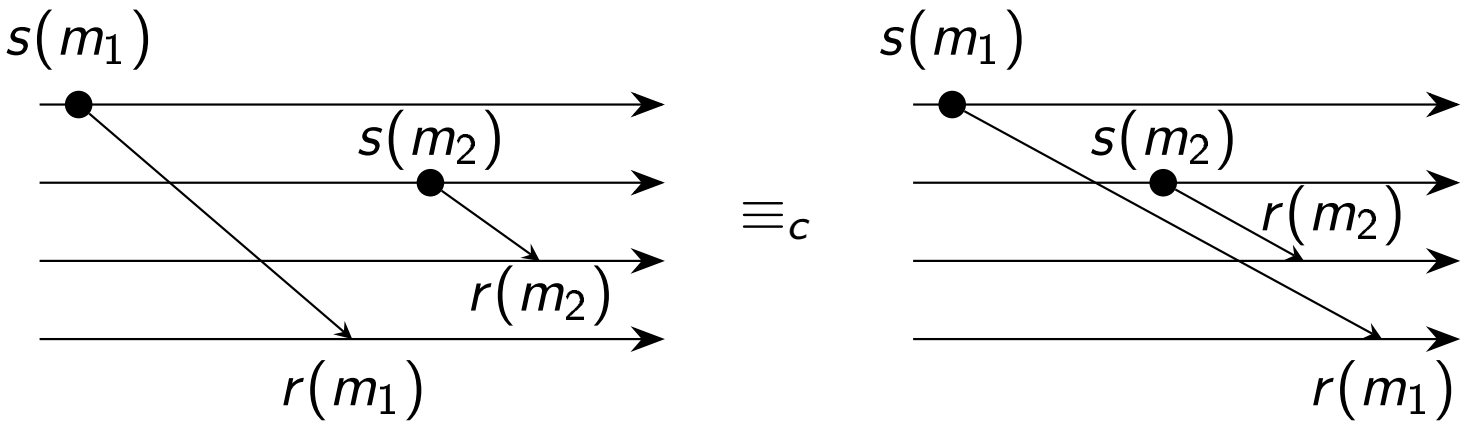
Our solution

For a predicate on executions, we define two predicates on computations:

- One captures detection in an asynchronous setting
- The other captures enforcement in an asynchronous setting

Table Of Contents

Causal Equivalence Classes



Causal Equivalence

For σ, σ' two executions, $\sigma \equiv_c \sigma' \triangleq$
σ and σ' are executions from the same computation.

Our Topology

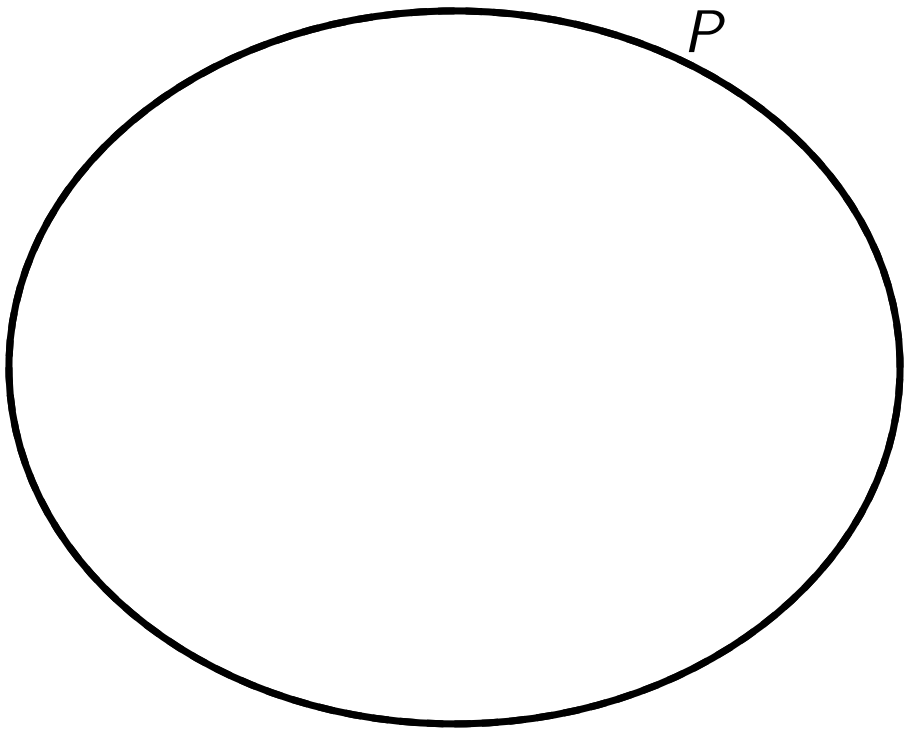
The Computation Topology

Let S a set of executions (equivalently, a predicate). Then S is an open set in the **Computation Topology** iff it is a union of causal equivalence classes.

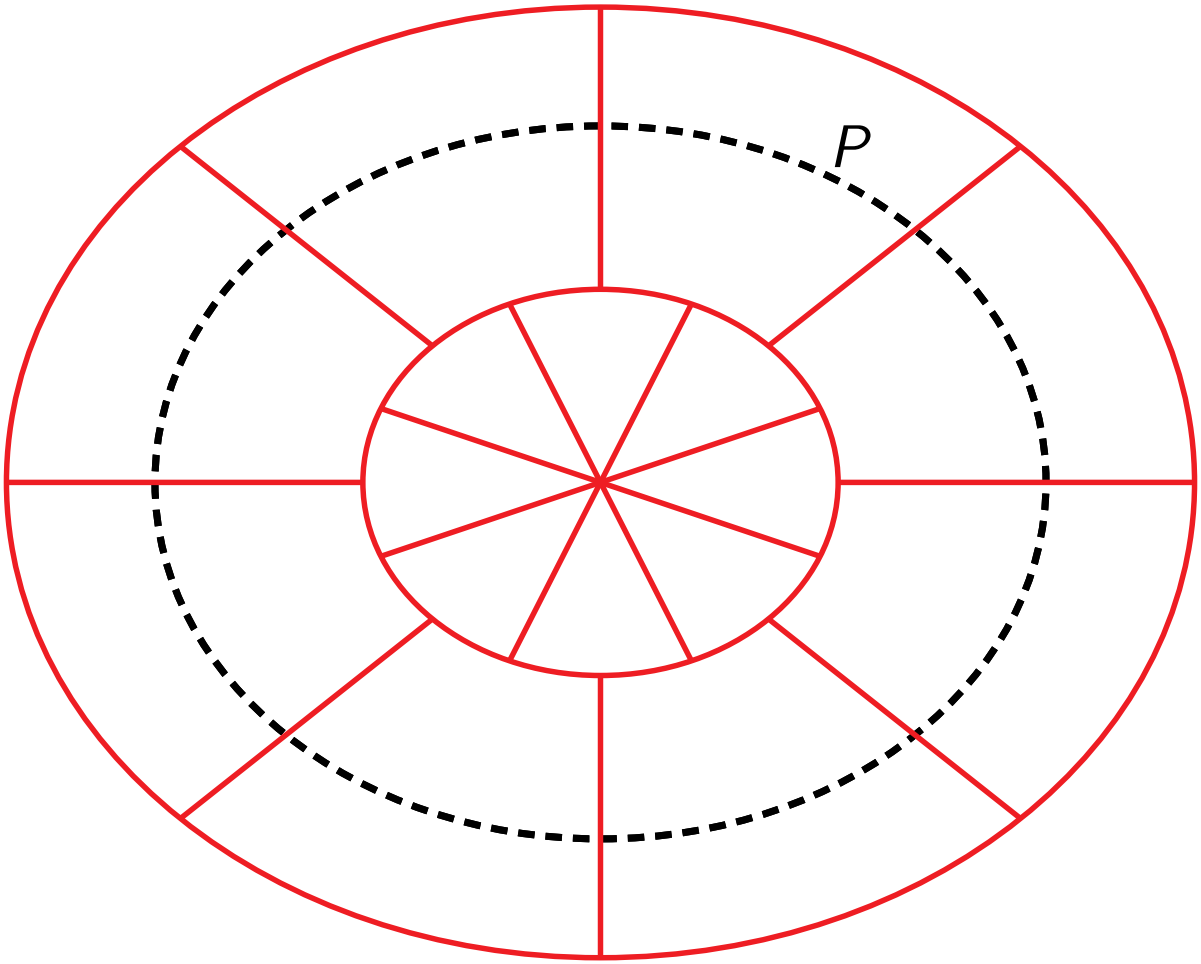
Some Properties of our Topology

- Open sets are isomorphic to sets of computations
- Every open set is also closed
- Every closed set is also open

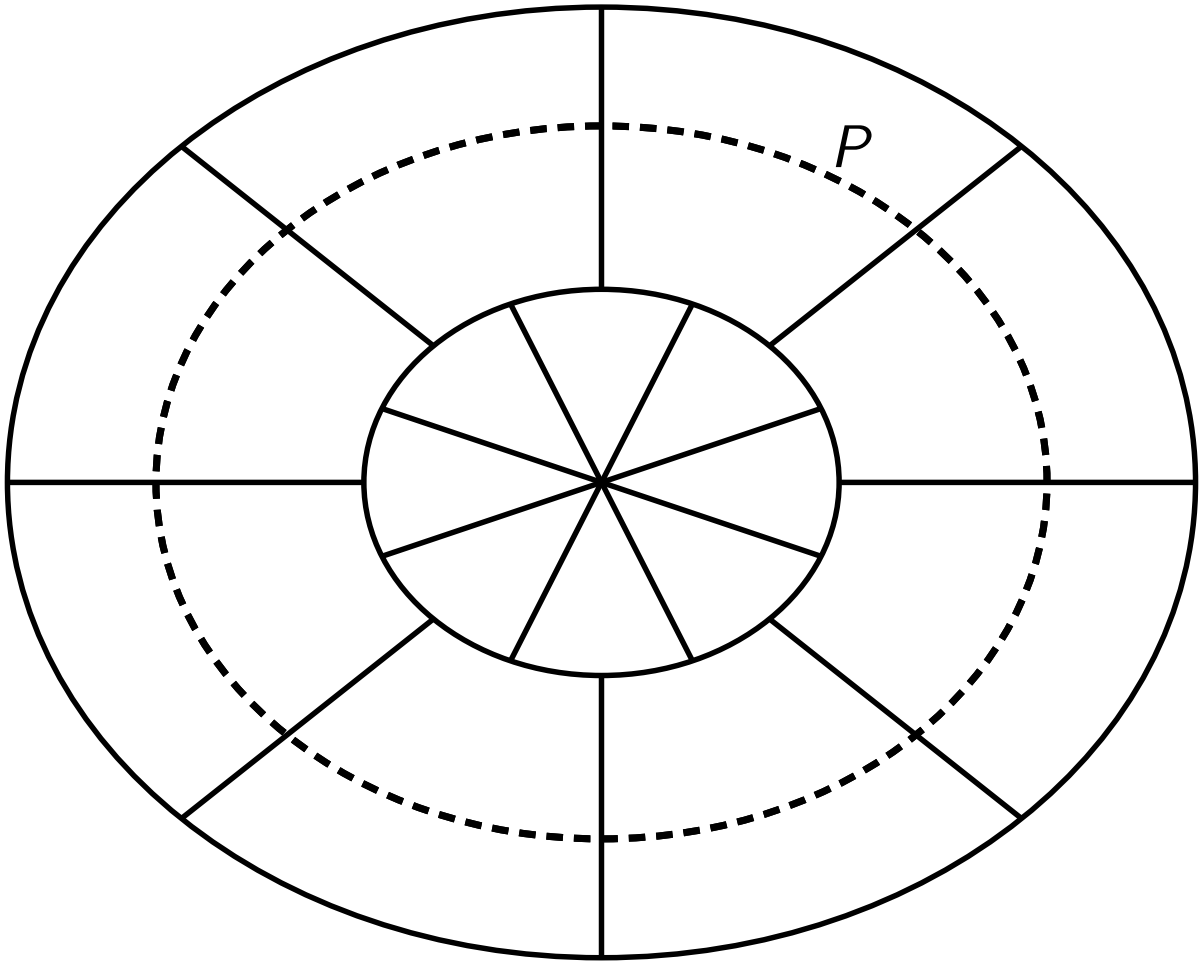
A Visual Formulation



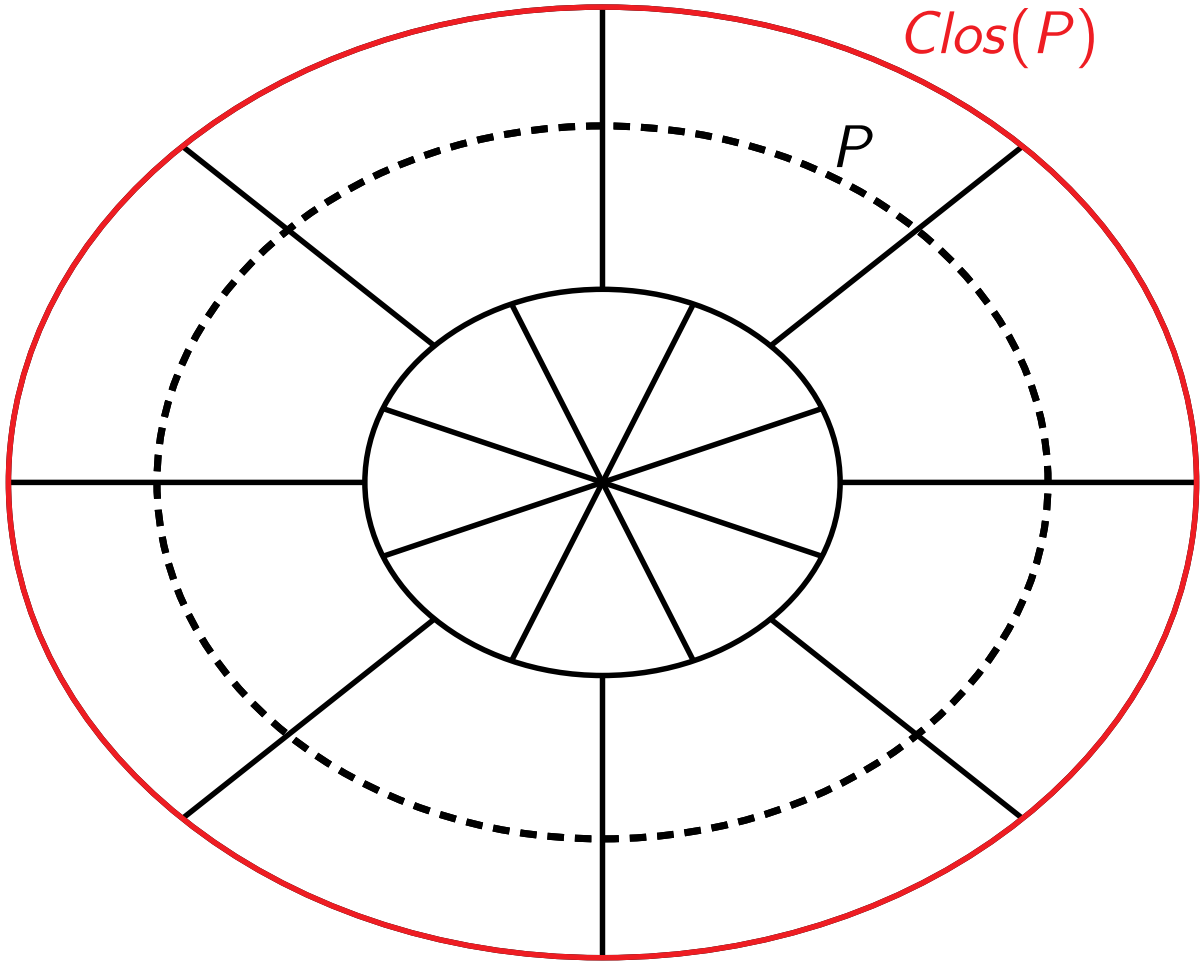
A Visual Formulation



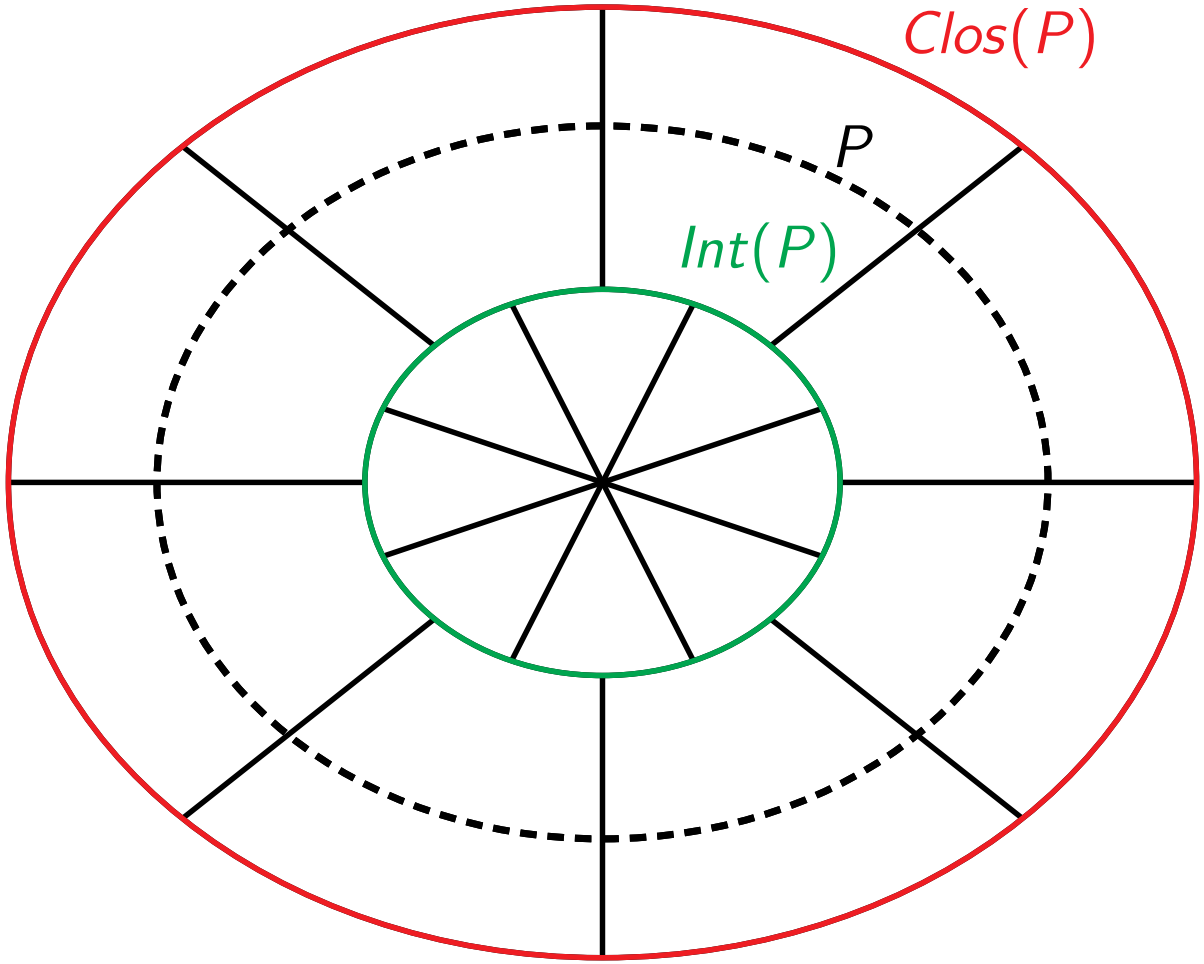
A Visual Formulation



A Visual Formulation



A Visual Formulation



Closures and Interiors

Closure

$Clos(P)$, the closure of P, is the smallest closed set such that $P \subseteq Clos(P)$.

It corresponds to the set of computations with at least one execution in P.

Closures and Interiors

Closure

$Clos(P)$, the closure of P, is the smallest closed set such that $P \subseteq Clos(P)$.

It corresponds to the set of computations with at least one execution in P.

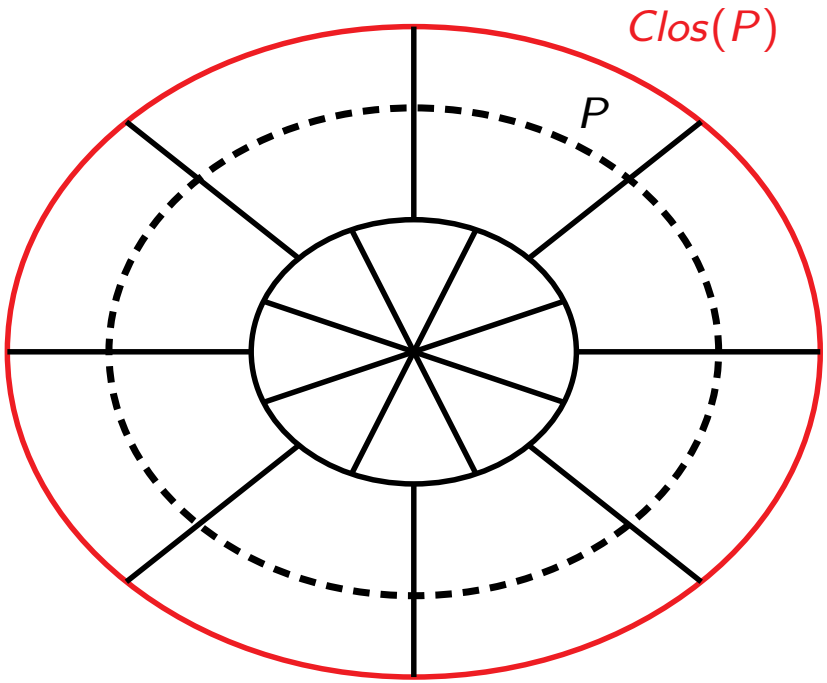
Interior

$Int(P)$, the interior of P, is the greatest open set such that $Int(P) \subseteq P$.

It corresponds to the set of computations with all executions in P.

Table Of Contents

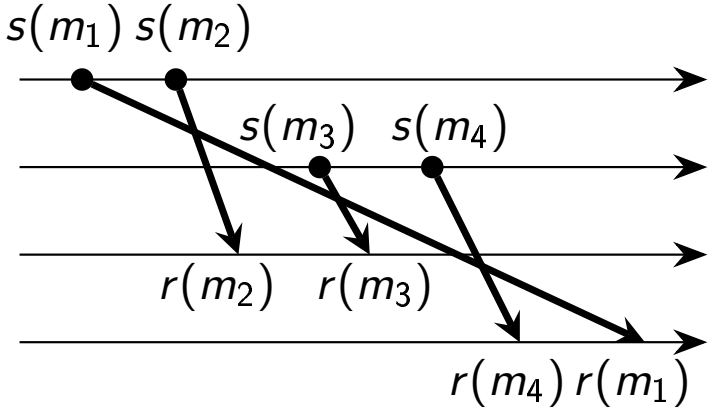
Closure as Overapproximation for Detection

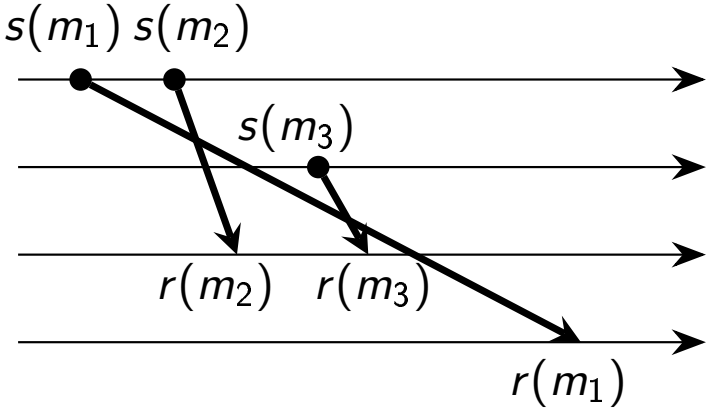


Interpretation

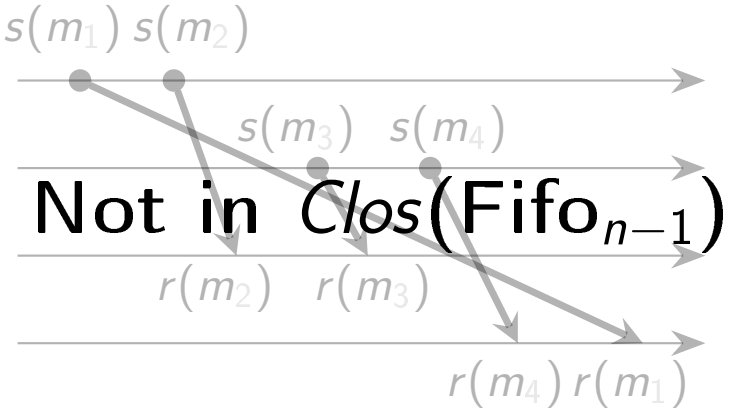
- Union of causal equivalence classes with at least one execution satisfying the predicate.
- Smallest overapproximation in an asynchronous setting.

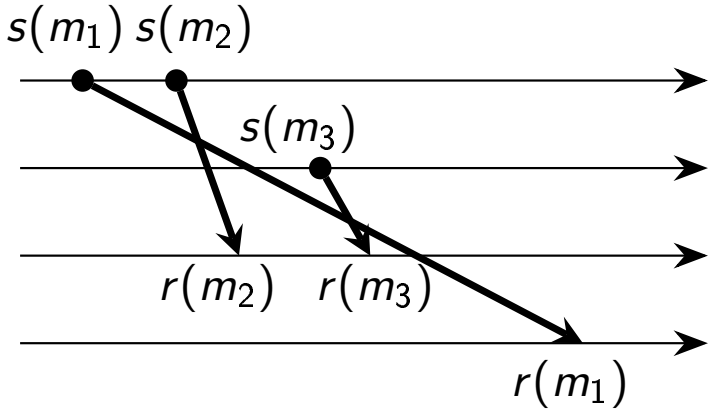
Overapproximating for Detection



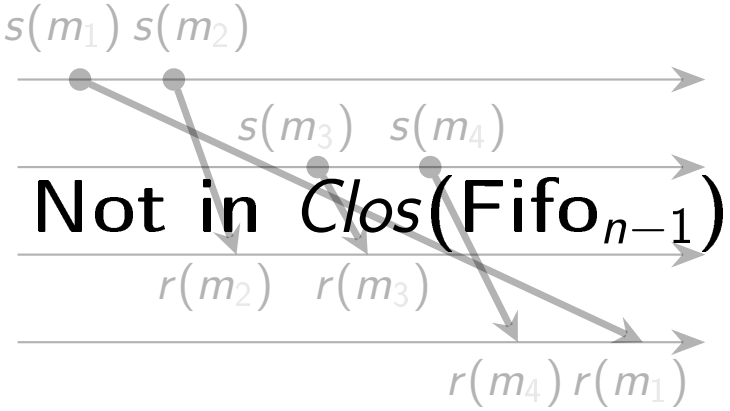


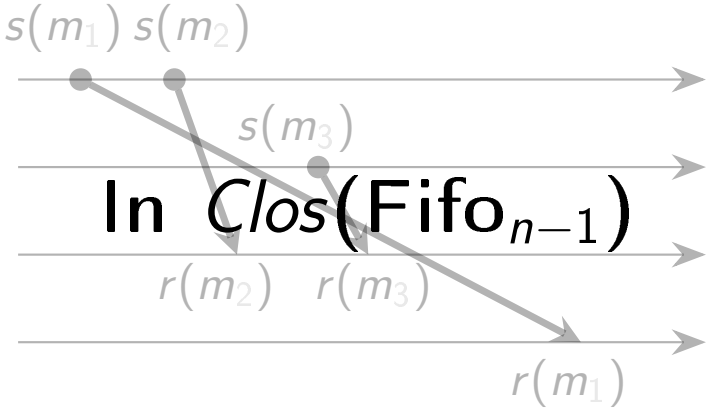
Overapproximating for Detection



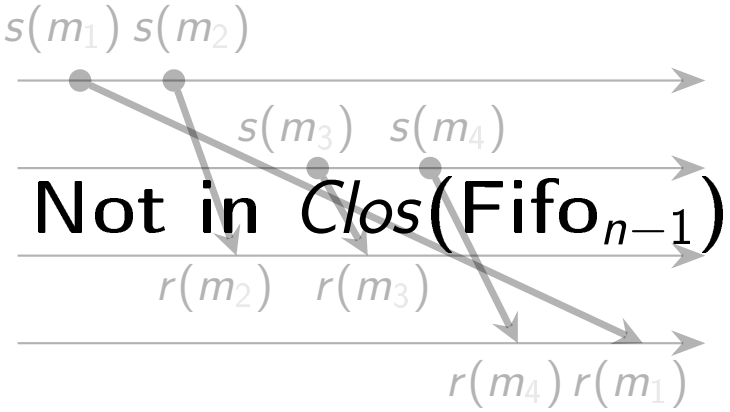


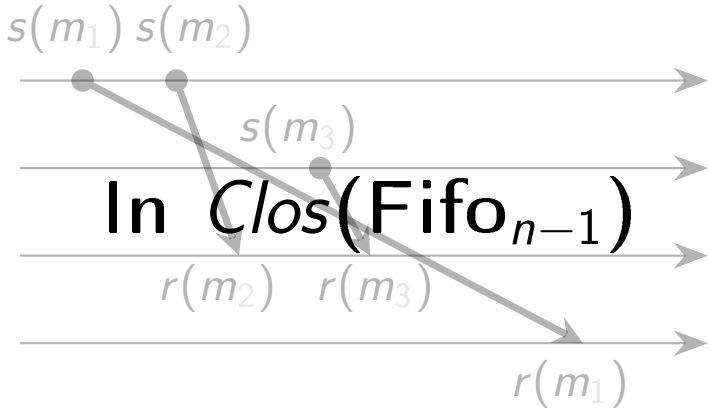
Overapproximating for Detection





Overapproximating for Detection





Closures as the smallest overapproximation for detection

One computation generated by the system is not in $Clos(\text{Fifo}_{n-1})$
$\Longrightarrow$ All executions of this computation are not in Fifo_{n-1}
$\Longrightarrow$ System is not Fifo_{n-1}.

Closure captures the smallest overapproximation in an asynchronous setting.

Characterization of Closures

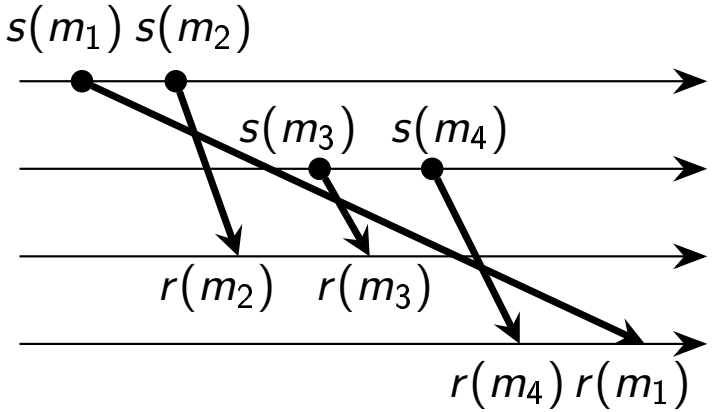
Forbidden patterns as characterizations of closures

We characterize the closures of specific delivery predicates by forbidden patterns in the causal order.

Characterization of Closures

Forbidden patterns as characterizations of closures

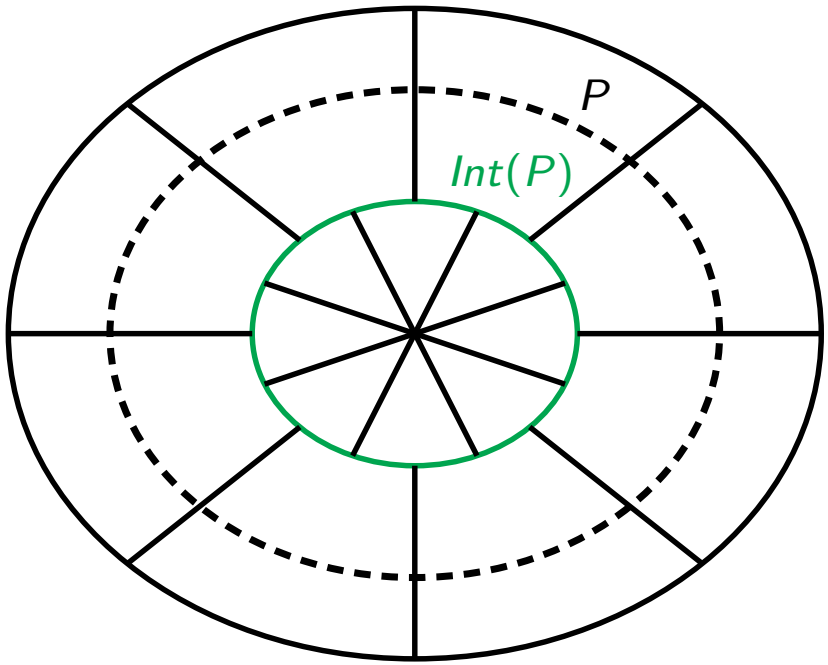
We characterize the closures of specific delivery predicates by forbidden patterns in the causal order.



$Clos(\text{Fifo}_{n-1})$ = no causal knots

For example, $Clos(\text{Fifo}_{n-1})$ is characterized by the absence of causal knots, patterns in the causal order.

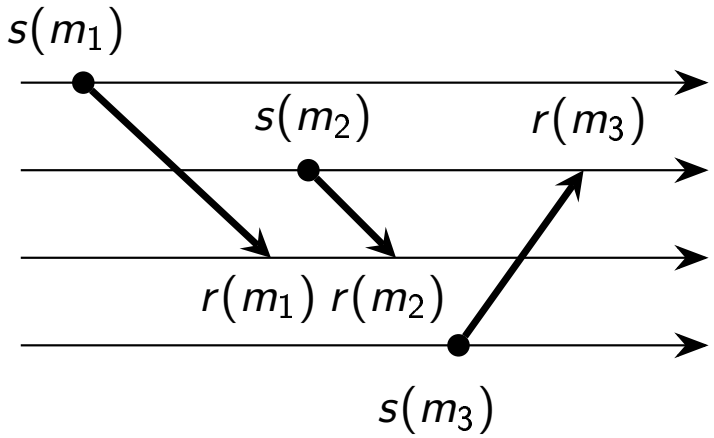
Interior as Underapproximation for Enforcement

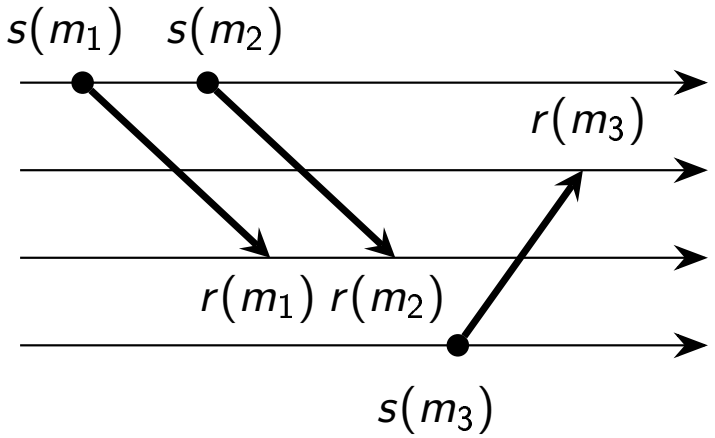


Interpretation

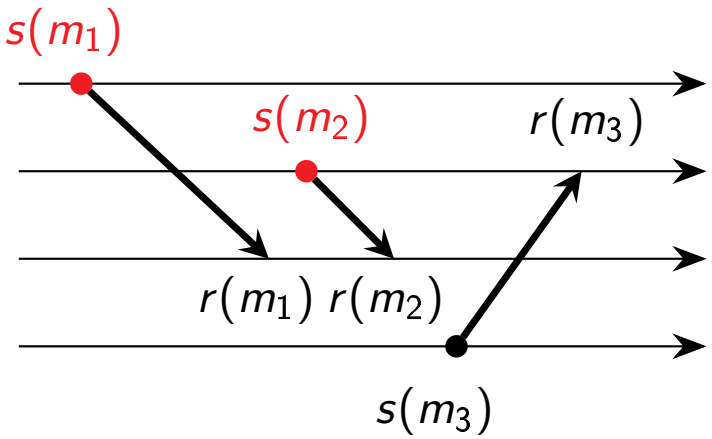
- Union of causal equivalence classes with all executions satisfying the predicate.
- Largest underapproximation in an asynchronous setting.

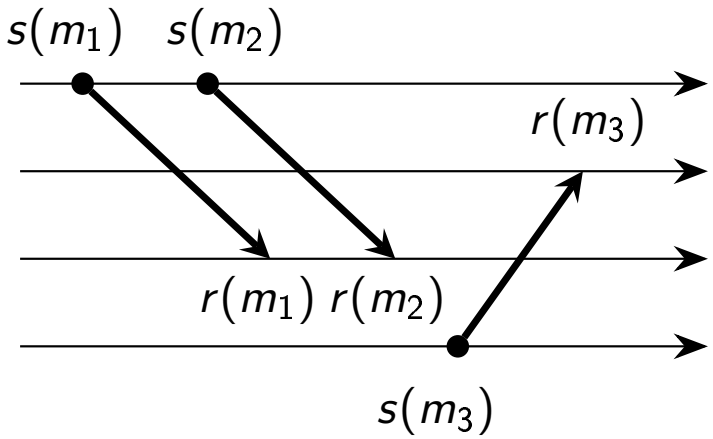
Underapproximating for Enforcement



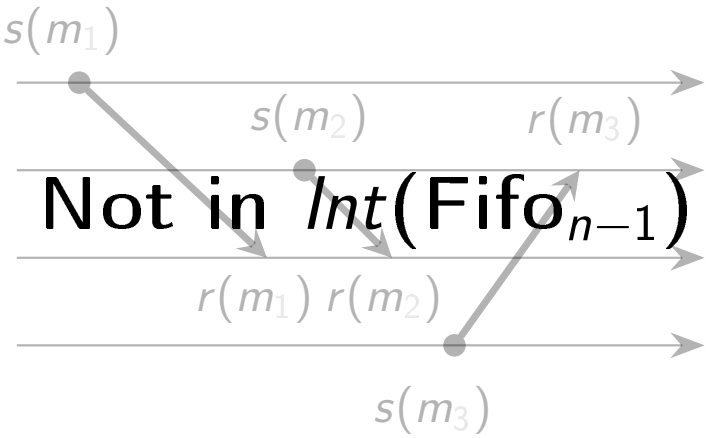


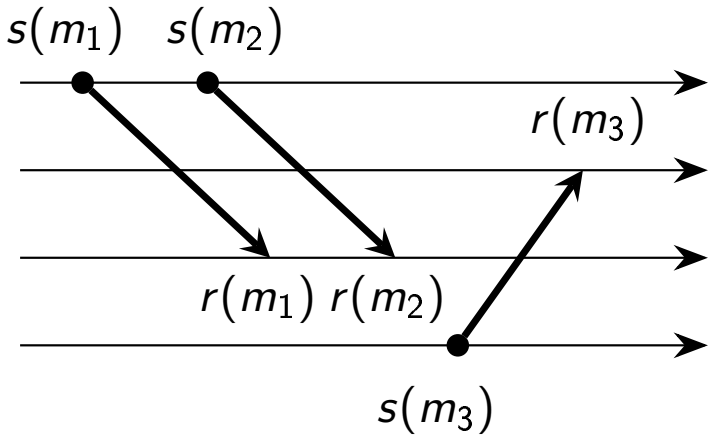
Underapproximating for Enforcement



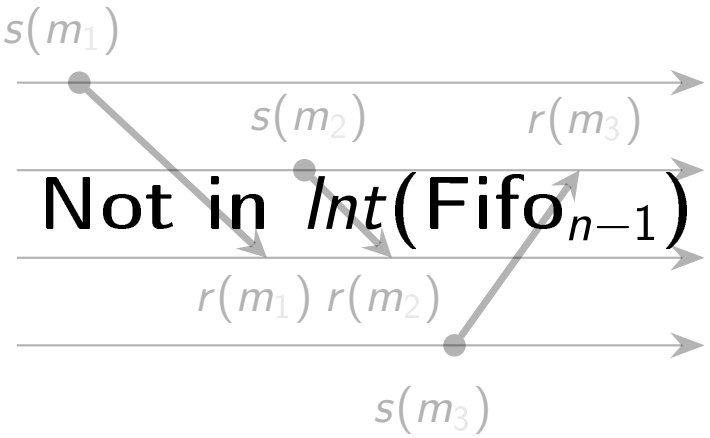


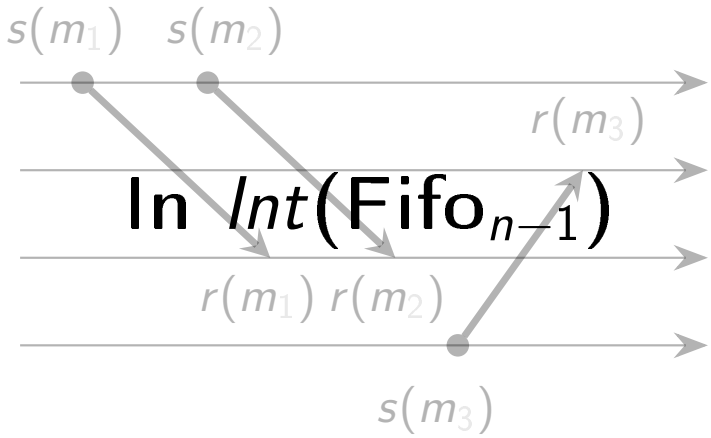
Underapproximating for Enforcement



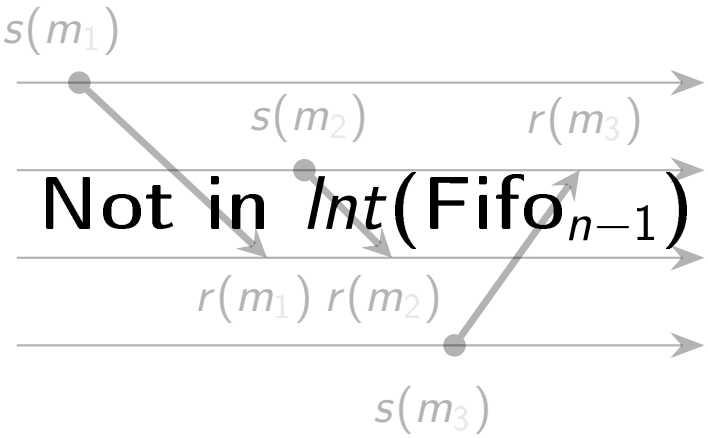


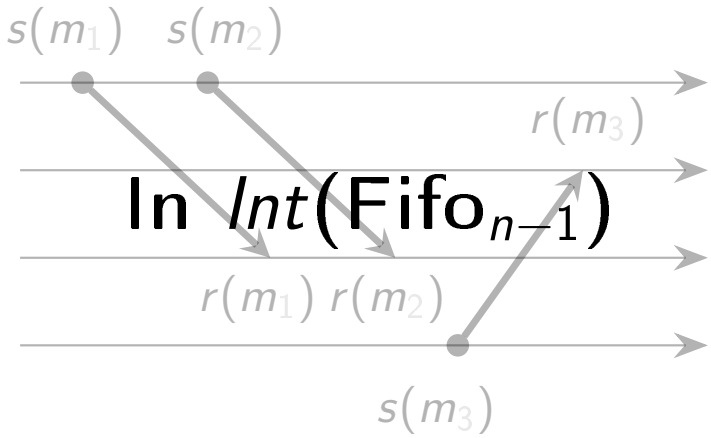
Underapproximating for Enforcement





Underapproximating for Enforcement



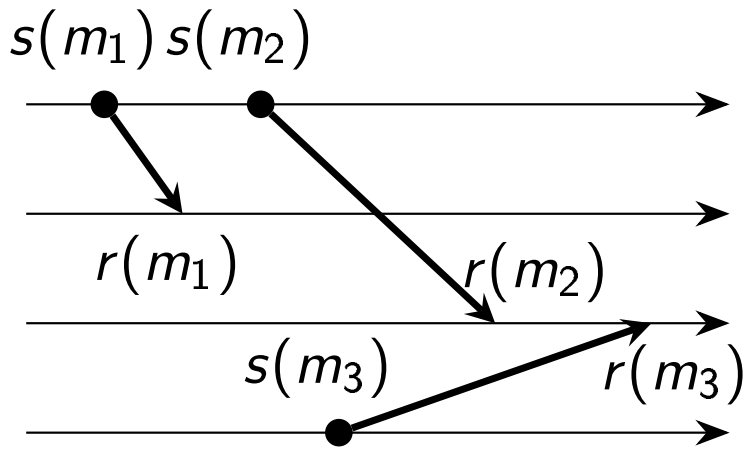


Interiors as means of Enforcement

All computations generated by the system are in $Int(\text{Fifo}_{n-1})$
$\Longrightarrow$ All executions generated by the system are in Fifo_{n-1}
$\Longrightarrow$ System enforces Fifo_{n-1}.

The interior is the largest underapproximation in an asynchronous world.

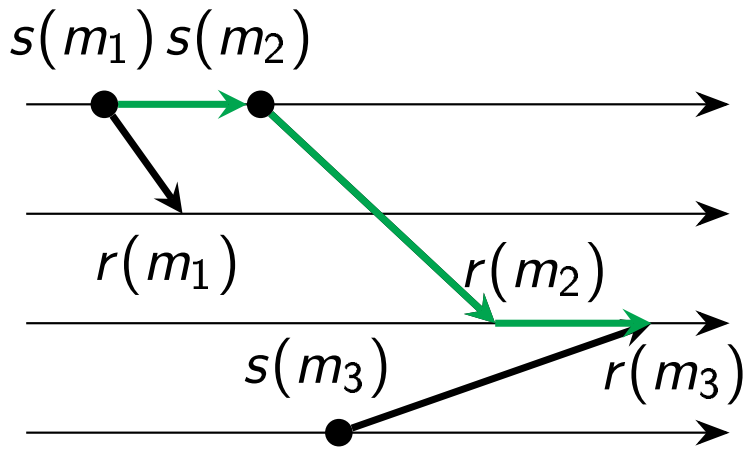
Characterization of Interiors



Chain

A **chain** C of a poset x is a totally ordered subset of x.

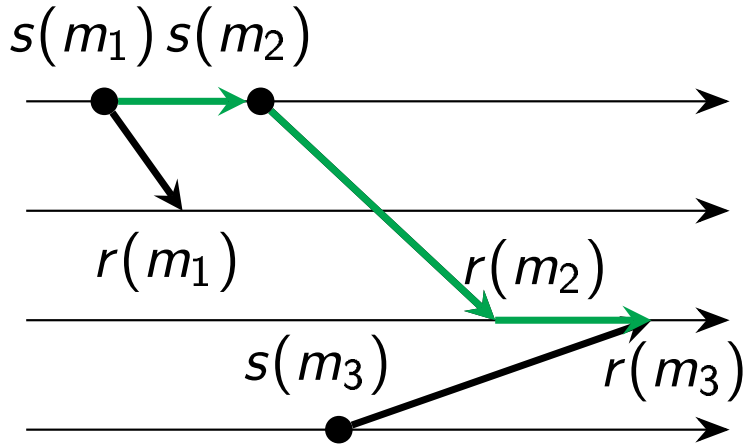
Characterization of Interiors



Chain

A **chain** C of a poset x is a totally ordered subset of x.
In a computation, a chain is a set of events totally ordered by causality.

Characterization of Interiors



Chain

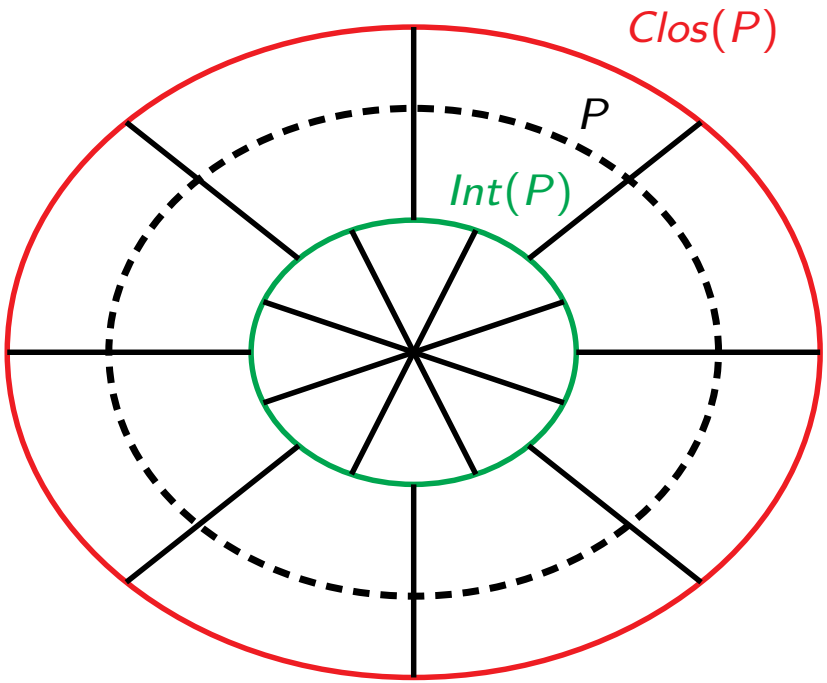
A **chain** C of a poset x is a totally ordered subset of x.
In a computation, a chain is a set of events totally ordered by causality.

Chains as characterizations of interiors

Chains allow us to encode a delivery predicate into the causal order, and thus to characterize Interiors.
Example : $Int(\text{Fifo}_{n-1})$ asks for all send events to the same receiving peer to form a chain.

Table Of Contents

Conclusion



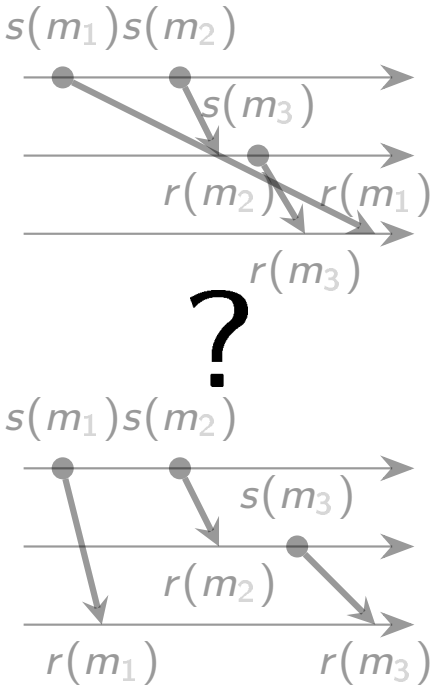
Summary

Through topology, we show the existence of two predicates on computations for any predicate on executions :

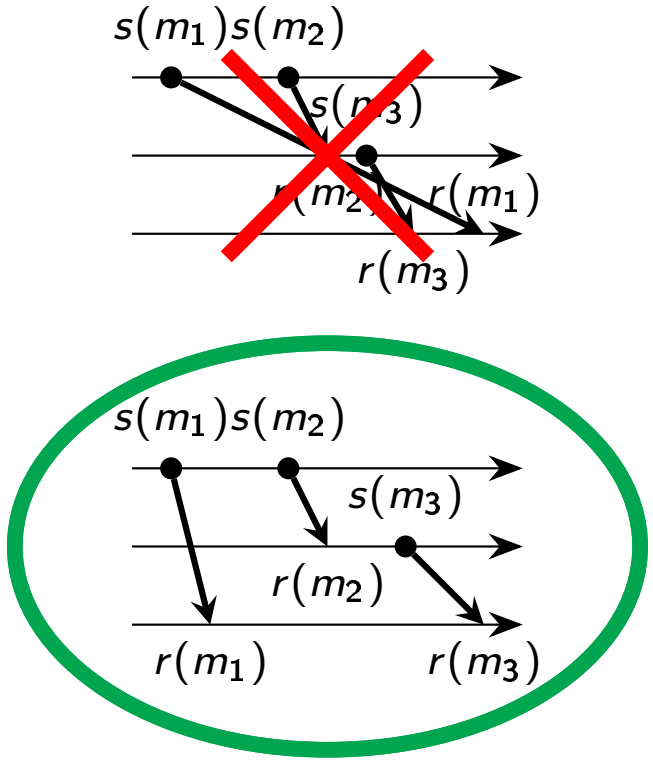
- its closure, the smallest overapproximation in the asynchronous setting
- its interior, the biggest underapproximation in the asynchronous setting

Back to our Two Questions

Detecting

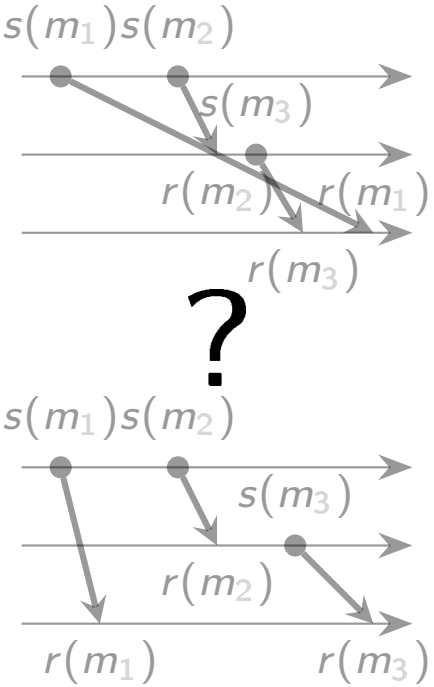


Enforcing

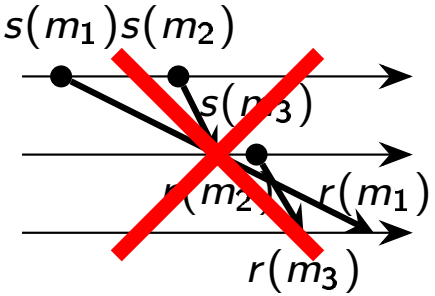


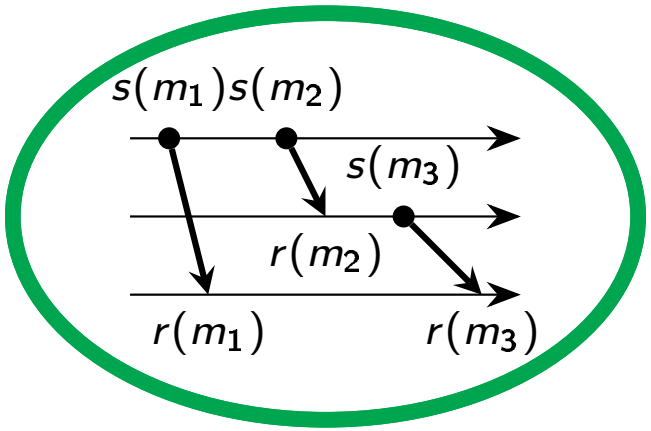
Back to our Two Questions

Detecting through Closures



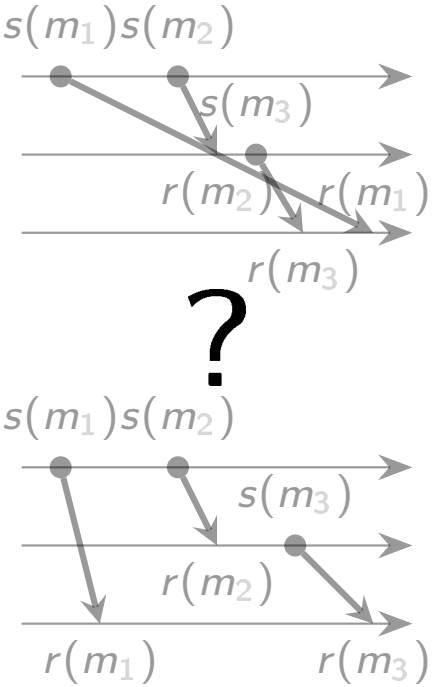
Enforcing



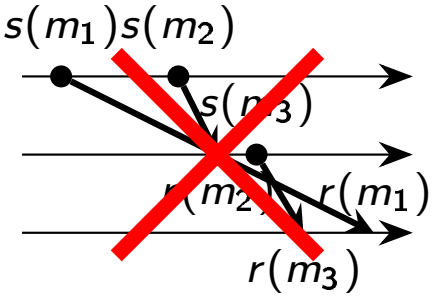


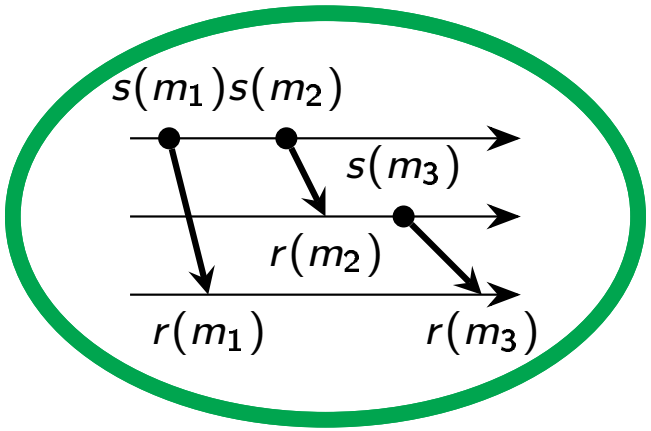
Back to our Two Questions

Detecting through Closures



Enforcing through Interiors





Perspectives

Study of other delivery predicates

We have characterized closures and interiors of point-to-point communication; we could also study multicast and broadcast delivery predicates.

Perspectives

Study of other delivery predicates

We have characterized closures and interiors of point-to-point communication; we could also study multicast and broadcast delivery predicates.

Study of non-delivery predicates

Characterizing the closures and interiors of arbitrary predicates on executions might also prove insightful.

Perspectives

Study of other delivery predicates

We have characterized closures and interiors of point-to-point communication; we could also study multicast and broadcast delivery predicates.

Study of non-delivery predicates

Characterizing the closures and interiors of arbitrary predicates on executions might also prove insightful.

The power given by predicates

Knowing that a predicate on executions is enforced provides additional knowledge about the ordering of events. What are the practical uses for this knowledge?