Scenario Automata

Theory and Applications

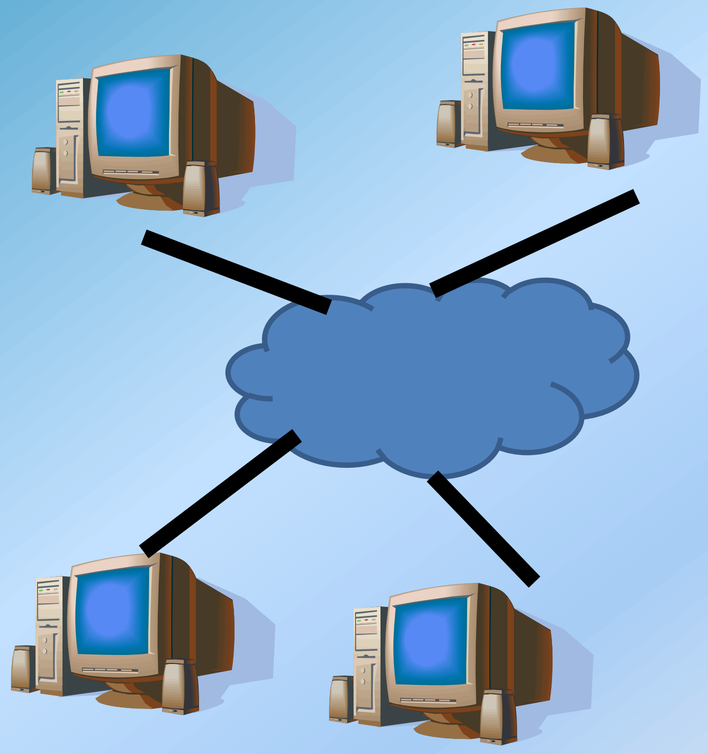
Jury:
Albert Benveniste
Roland Groz
Paul Gastin
Claude Jard
Martin Leucker
Madhavan Mukund
Sophie Pinchinat
P.S. Thiagarajan

Loïc Hélouët

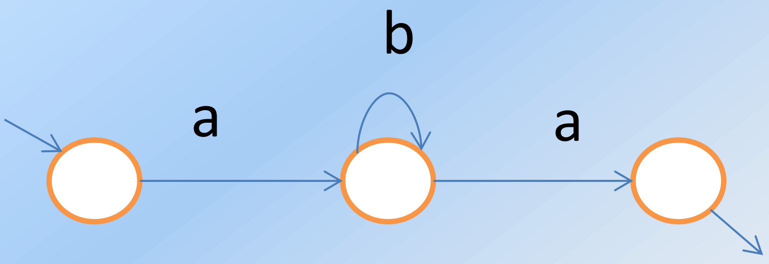
May 17th 2013

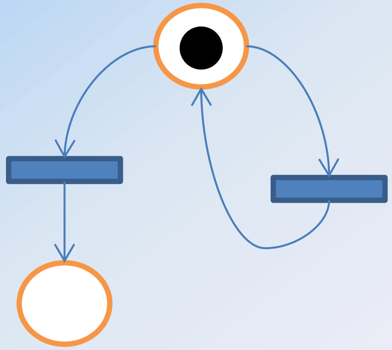
Motivations

Real World



Models





Motivations

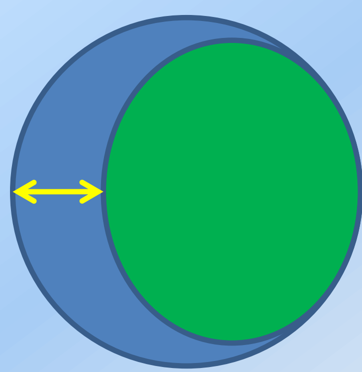
Real World

A network of Turing Machines

Highly undecidable

Needs:

Design, validation, monitoring,…



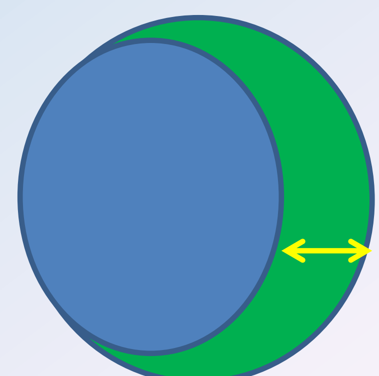
Underapproximation

Models

- Automated techniques
- Decidable properties
- Clear Semantics

But:

Close enough to real world ?



Overapproximation

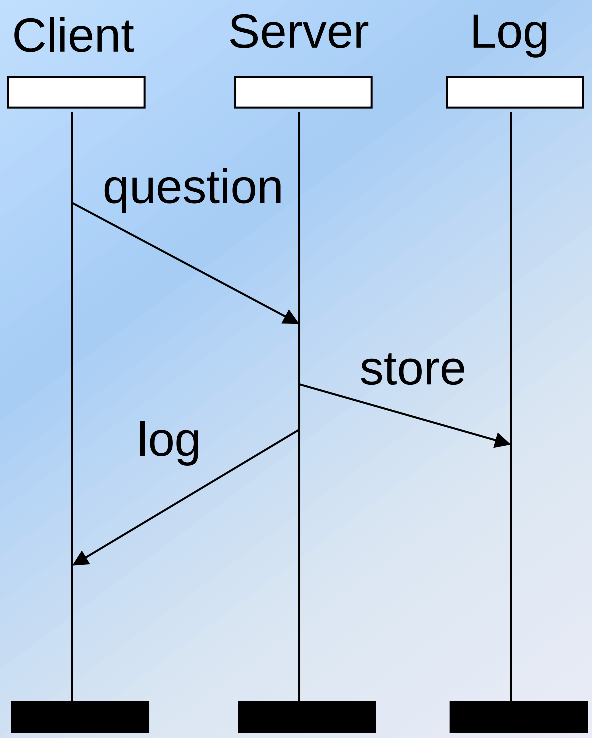
Motivations

Why scenarios / partial order models ?

- Intuitive
- Compact representation of concurrency

Objectives: tools for engineers

- Models for asynchronous & distributed systems behaviors
- Automated verification/analysis
- *Avoiding global states computation*



An example MSC

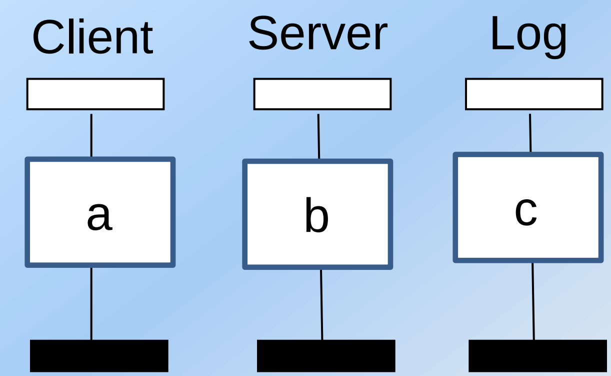
Motivations

Why scenarios / partial order models ?

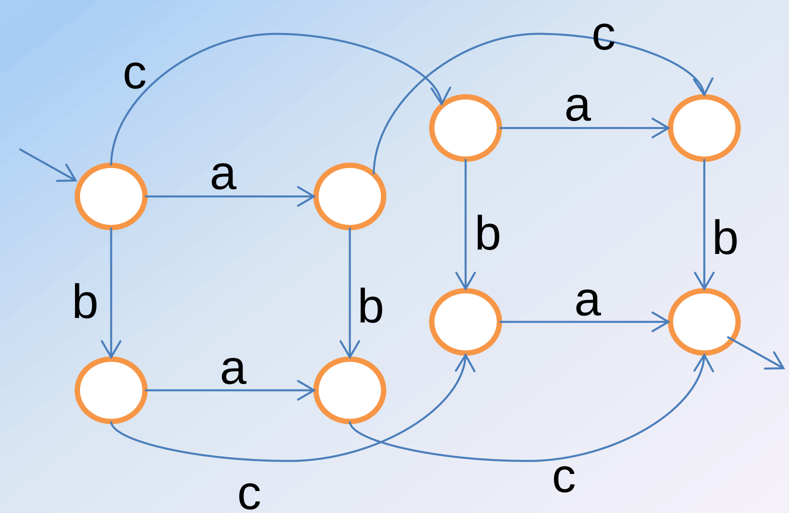
- Intuitive
- Compact representation of concurrency

<u>Objectives</u>: tools for engineers

- Models for asynchronous & distributed systems behaviors
- Automated verification/analysis
- *Avoiding global states computation*



MSC : 3 events



Automaton : 12 transitions

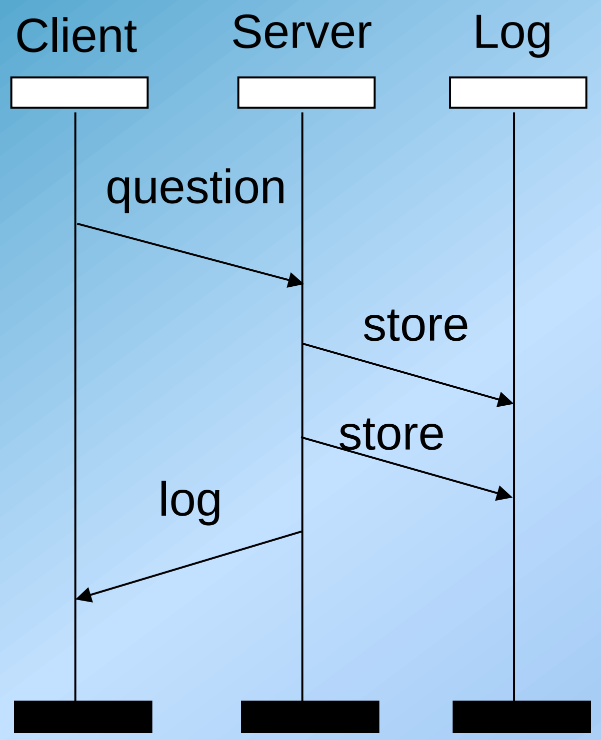
Outline

- Scenario automata
- Extensions
- Operators
- Verification & partial order logics
- Application
- Conclusion

Outline

- Scenario automata
- Extensions
- Operators
- Verification & partial order logics
- Application
- Conclusion

Scenario automata : MSCs



$M=(E, \{<_p\}_{p \in P}, \alpha, \mu, \phi)$

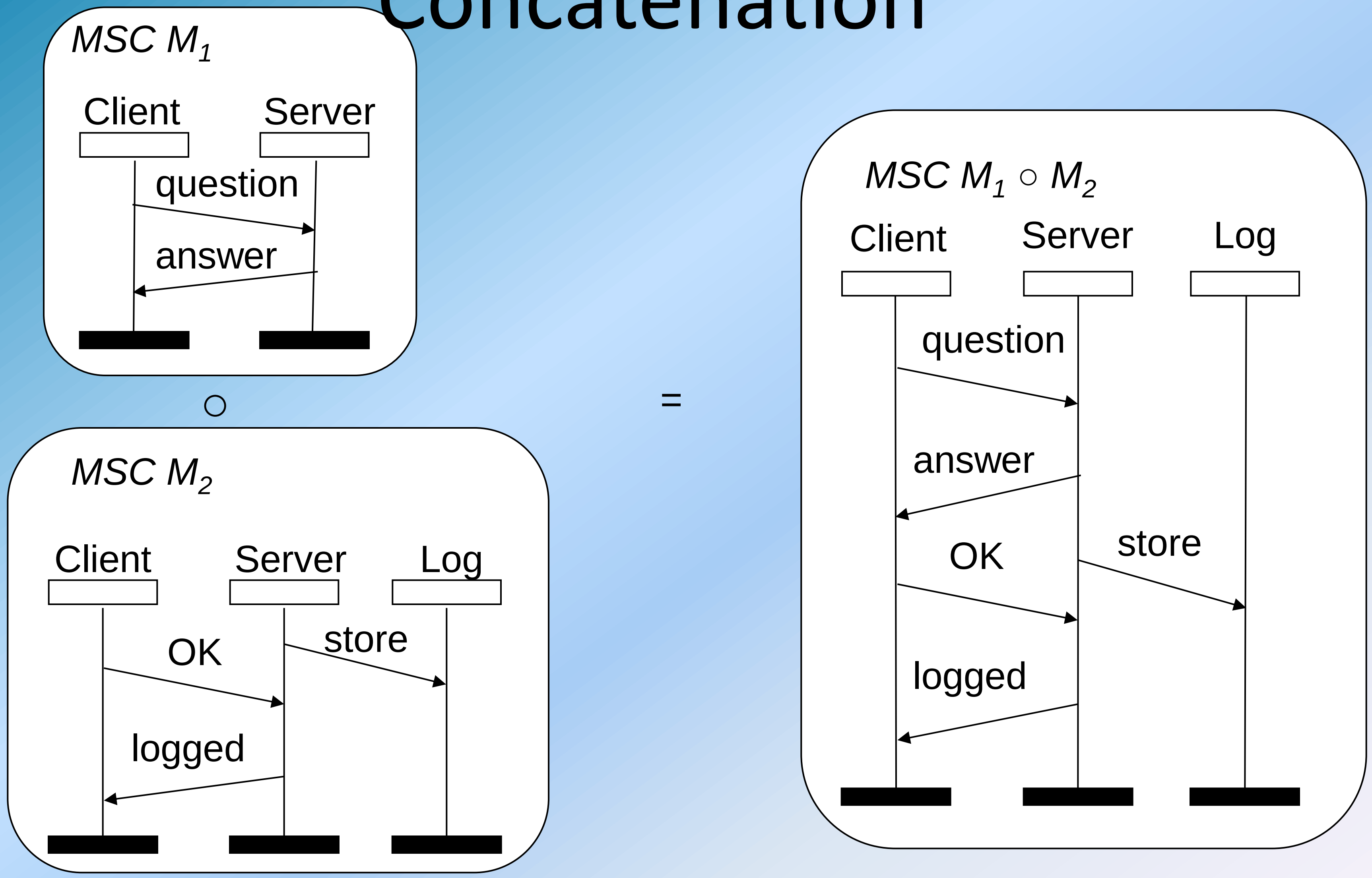
Labelled partial order over a finite set of events E

- P : set of processes
- $\alpha \rightarrow \Sigma$: labeling : α(e)=Client !Server(question), …
- $\phi : E \rightarrow P$: locality of events
- $<_p$: total ordering for each process $p \in P$
- μ : message pairing

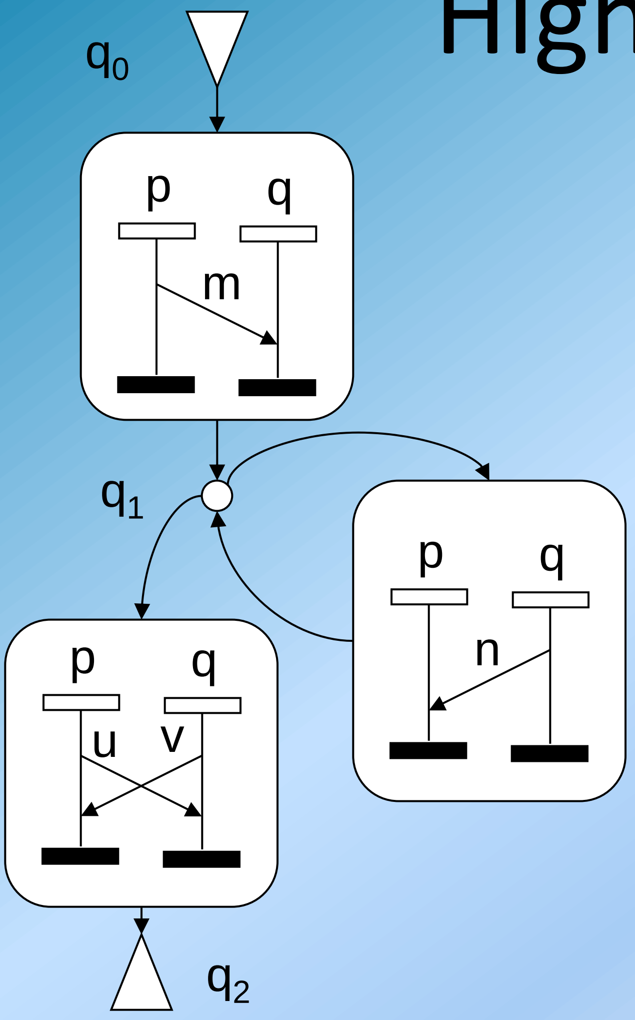
$$\left(\bigcup_{p \in P} <_p \cup \mu \right)^*$$ must be a partial order

Lin(M) = linearizations of $\bigcup_{p \in P} <_p \cup \mu$

Concatenation



High-level MSCs



An example HMSC H

$H=(Q,\rightarrow,M,q_0,Q_{Fi})$

A scenario automaton defined over a finite set of MSCs M

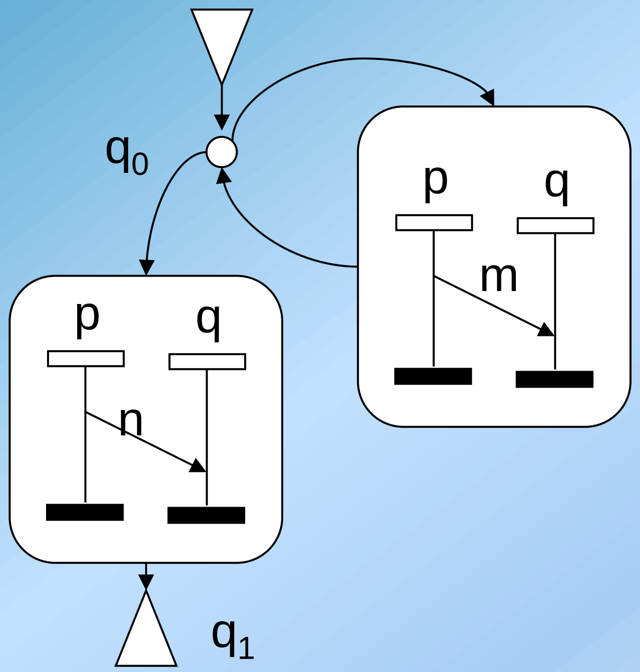
Path $\rho = q_0 \xrightarrow{M1} q_1 \xrightarrow{M2} \ldots \xrightarrow{Mk} q_k$

$\rho^{\circ} = M_1 \circ M_2 \circ \ldots M_k$

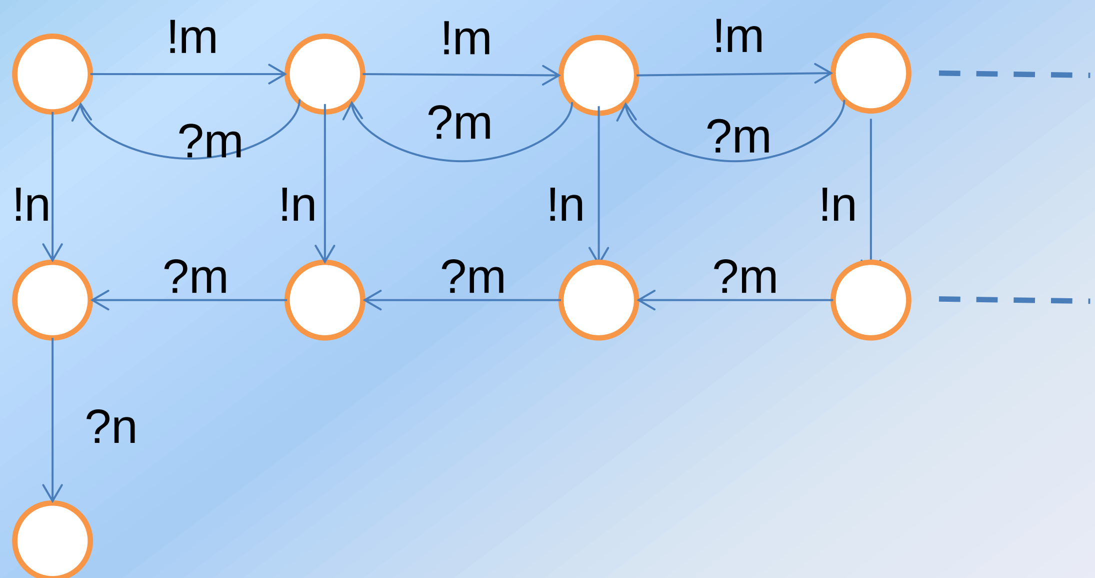
Semantics

- $P_H = \{ \rho = q_0 \xrightarrow{M1} q_1 \xrightarrow{M2} \ldots \xrightarrow{Mk} q_k \mid q_k \in Q_{Fi} \}$
- $F_H = \{ \rho^{\circ} \mid \rho \in P_H \}$
- $Lin_H = \bigcup_{M \in F_H} Lin(M)$

High-level MSCs



An HMSC



An equivalent transition system
…with inifite state space

Undecidable problems

Let H_1, H_2 be HMSCs

$F_{H_1} \cap F_{H_2} = \emptyset$? — $Lin_{H_1} \cap Lin_{H_2} = \emptyset$? [Muscholl et al 99]

$F_{H_1} \subseteq F_{H_2}$? — $Lin_{H_1} \subseteq Lin_{H_2}$? [Darondeau et al 00]

$F_{H_1} = F_{H_2}$? — $Lin_{H_1} = Lin_{H_2}$?

Lin_{H_1} Regular ? [Henriksen et al 05]

Let R be a regular subset of Σ^* :

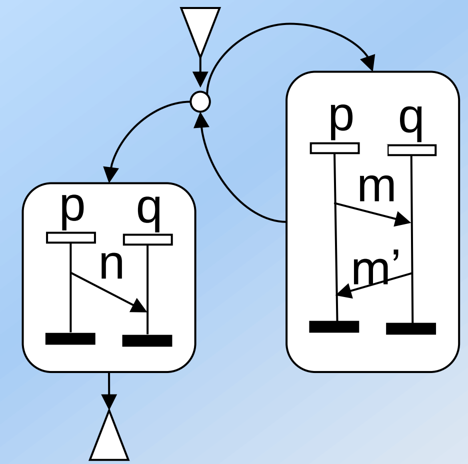
$R \subseteq Lin_{H_1}$?

$Lin_{H_1} \subseteq R$?

[Alur et al 99]

Not surprising: HMSCs closely related to Mazurkiewicz traces

Regular HMSCs [Alur et al 99] [Muscholl et al 99]



An HMSC H is **regular** iff for every cycle ρ *of H* $(\phi(E_{\rho^\circ}), \phi(\mu_{\rho^\circ}))$ is a *strongly connected graph*

every message sent is acknowledged after a finite nb of iterations of ρ*.*

<u>*Theorem:*</u>

H regular $\Rightarrow$ Lin_H regular subset of Σ^*

<u>Consequences:</u>

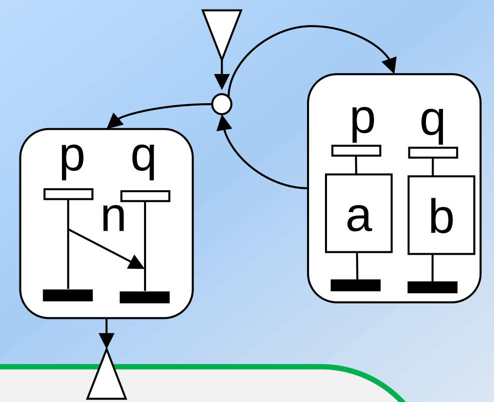
$Lin_{H1} \cap Lin_{H2} = \emptyset$, $Lin_{H1} \subseteq Lin_{H2}$, $Lin_{H1} = Lin_{H2}$

$R \subseteq Lin_{H1}$, $Lin_{H1} \subseteq R$ decidable

Globally cooperative HMSCs

[Genest et al 02][Morin02]

An HMSC H is **globally cooperative** iff for every cycle ρ *of H* $(\phi(E_{\rho^\circ}), \phi(\mu_{\rho^\circ}))$ *is a connected graph*



Processes do not behave independently in a loop

<u>Theorem:</u> [Genest et al 02]

Let H_1 be a HMSC,
H_2 be a globally cooperative HMSC, then

$F_{H_1} \cap F_{H_2} = \emptyset$?
$F_{H_1} \subseteq F_{H_2}$? are decidable

There is a bound $b \in N$ s.t all $M \in F_H$ can be run with communication buffers of size $\leq b$. (Existential bound)

Outline

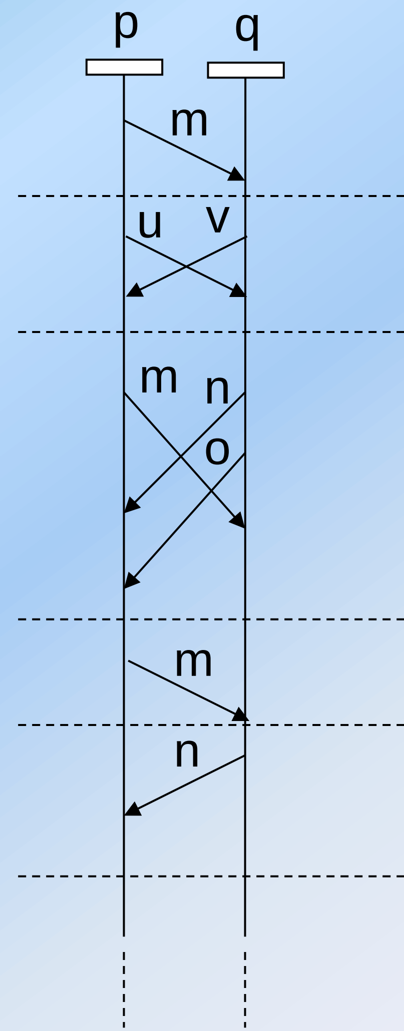
- Scenario automata
- Extensions
  - Compositional MSCs
  - Causal HMSCs
  - Dynamic MSC Grammars
- Operators
- Verification & partial order logics
- Application
- Conclusion

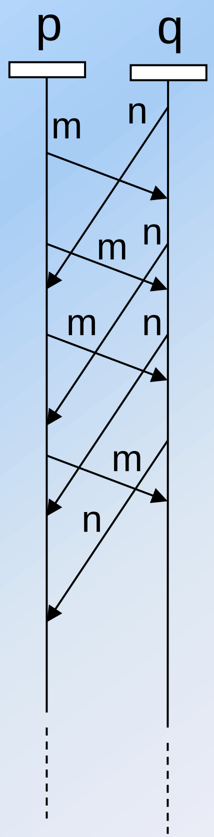
A drawback of HMSCs

MSC languages generated by

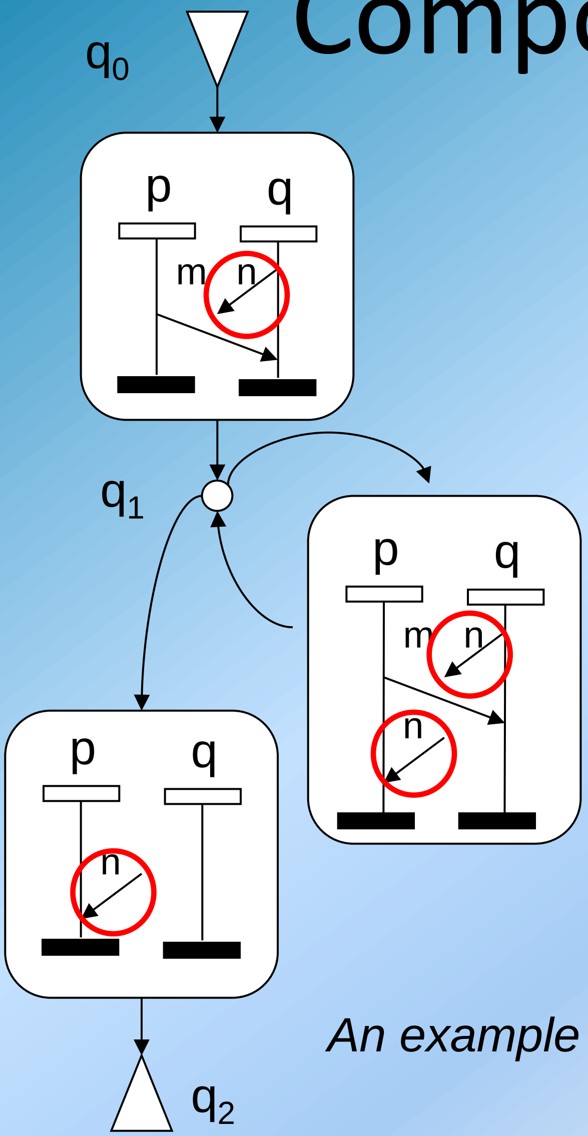
- HMSCs,
- GC-HMSCs,
- regular HMSCs …

…are all finitely generated





Compositional HMSCs



[Gunter et al'01, GenestPhD04]

$H=(Q,\to,M,q_0,Q_{Fi})$

A partial order automaton defined over a set of cMSCs M

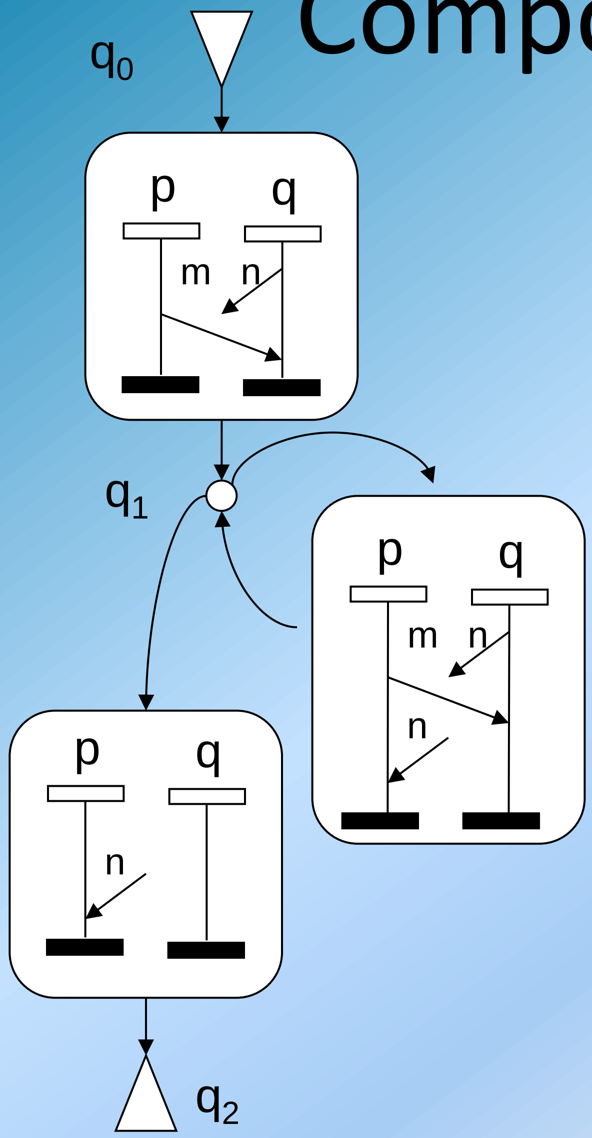
Dangling Messages emissions/receptions
Glued at composition time

Semantics:

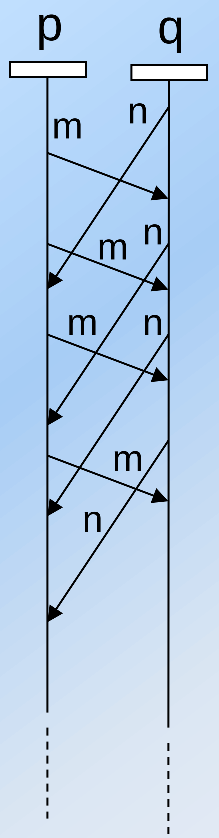
- $P_H = \{\rho = q_0 \xrightarrow{M1} q_1 \xrightarrow{M2} \ldots \xrightarrow{Mk} q_k \mid q_k \in Q_{Fi}\}$
- $F_H = \{\rho^\circ \mid \rho \in P_H$ and ρ° *is an MSC* $\}$
- $Lin_H = \bigcup_{M \in F_H} Lin(M)$

An example C-HMSC

Compositional HMSCs



Generates



Clearly not finitely generated !

Undecidable problems

C-HMSCs embed :

- HMSCs
- Communicating finite state machines (CFSM)

… and all their undecidable problems

<u>The simple message problem:</u>

is there an MSC $M \in F_H$ that contains a message of type m ?

<u>The emptiness problem:</u>

is F_H empty ?

are undecidable

Subclasses of cHMSCs

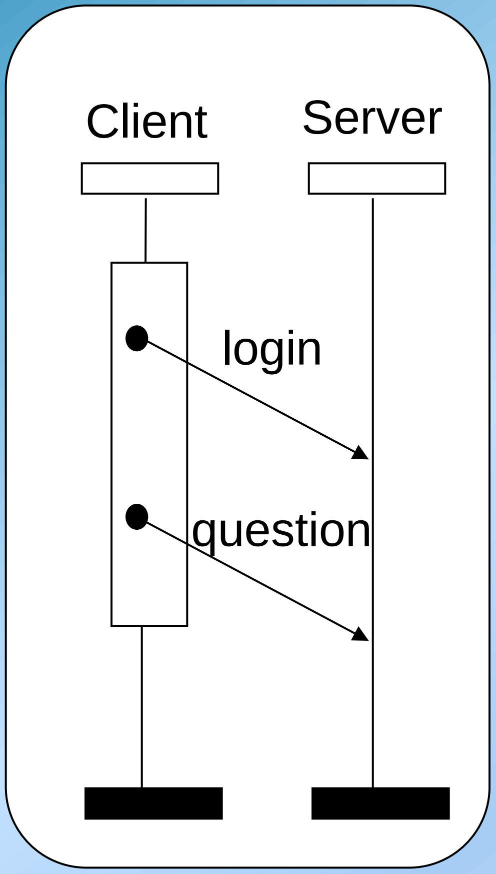
Some paths of a cHMSC may not generate an MSC

- A cHMSC H is safe is for every path ρ of P_H, ρ° is an MSC.

  (safe CHMSCs do not embed the whole expressive power of CFSMs)

- Globally cooperative CHMSCs = safe + GC
- Regular CHMSCs = safe + regular

Causal MSCs [GGHTY07, GGHTY09]



Labelled partial order over a set of events

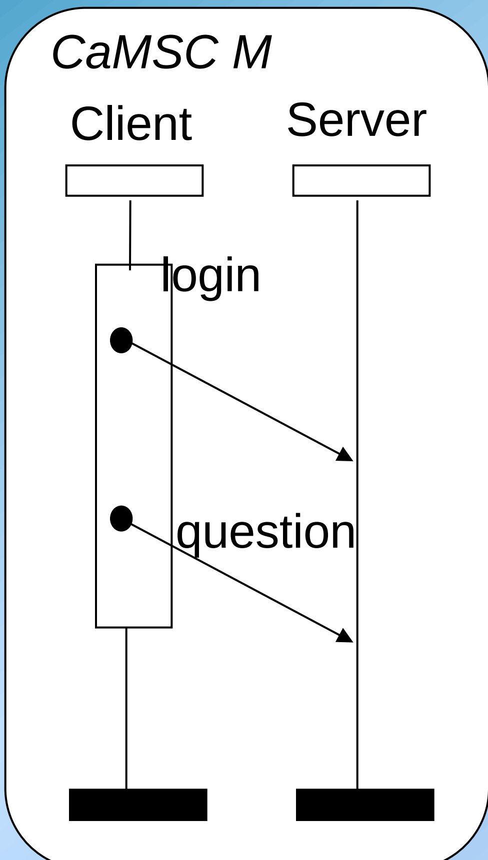
$M = (E\ ,\ \{\sqsubseteq_p\}_{p\in P}\ ,\ \alpha,\ \mu,\ \phi\ )$

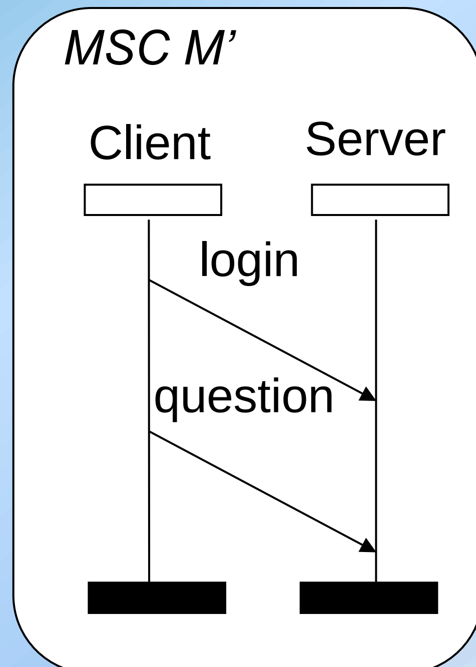
- $P,\ \alpha, \phi$ as usual,
- μ : message pairing as usual
- $\sqsubseteq_p$: partial order

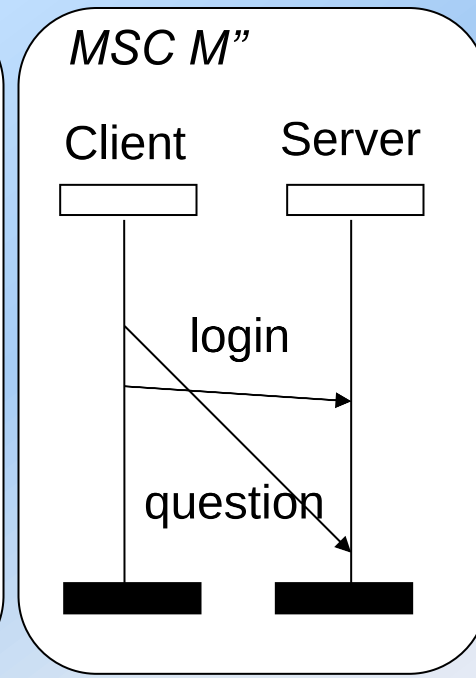
$$\Big(\bigcup_{p\in P} \sqsubseteq_p \cup\, \mu\Big)^*$$ is a partial order

Lin(M) = linearizations of $\bigcup_{p\in P} \sqsubseteq_p \cup\, \mu$

Visual extensions





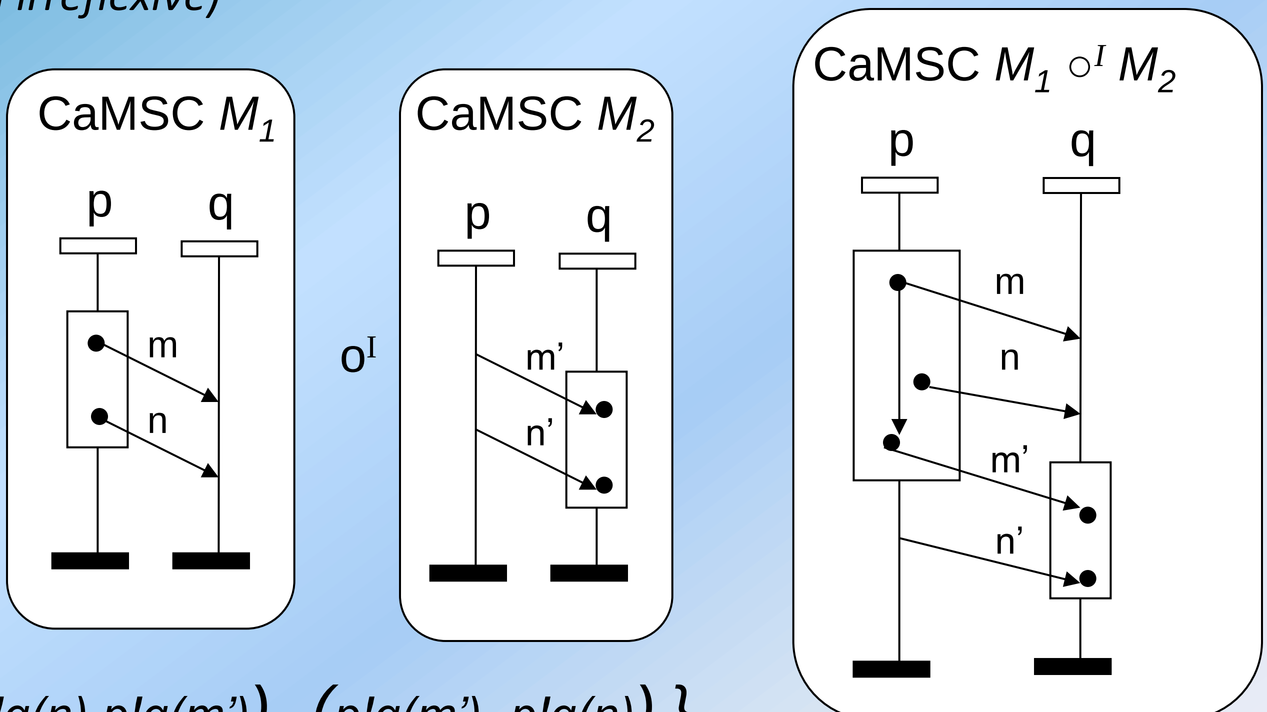


Vis(*M*) = MSCs that are compatible with *M*

Concatenation

<u>Independence relation</u> $I_p \subseteq \Sigma_p \times \Sigma_p$ for each $p \in P$

(Symmetric and irreflexive)

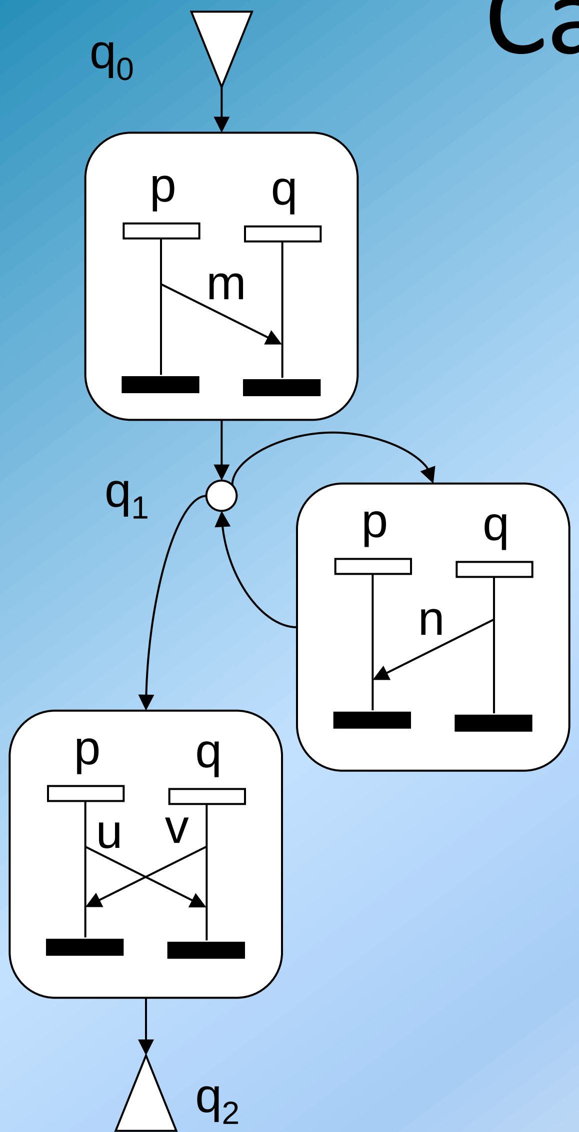


$I_p = \{ (p!q(n), p!q(m')) , (p!q(m'), p!q(n)) \}$

$I_q = \varnothing$

Note: M_1 or M_2 need not respect $\{ I_p \}_{p \in \mathbf{P}}$

Causal HMSCs



$H=(Q,\rightarrow,M,q_0,Q_{Fi})$
$+ \ \{ I_p \}_{p \in P}$

A partial order automaton defined over a finite set of Causal MSCs M

Path $\rho=q_0 \xrightarrow{M1} q_1 \xrightarrow{M2} \dots \xrightarrow{Mk} q_k$

$\rho^{\circ l} = M_1 \circ^l M_2 \circ^l \dots M_k$

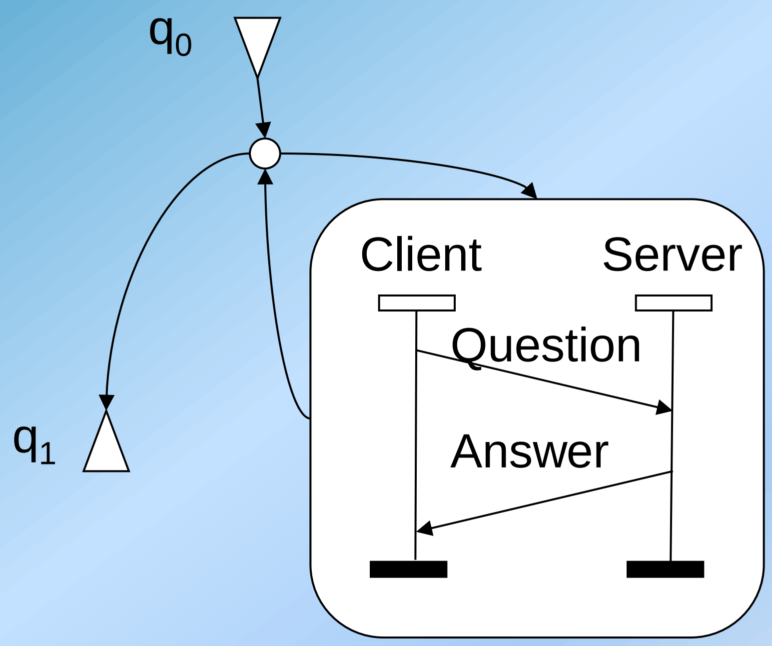
Semantics:

- $P_H =\{\rho=q_0 \xrightarrow{M1} q_1 \xrightarrow{M2} \dots \xrightarrow{Mk} q_k \mid q_k \in Q_{Fi}\}$
- $F_H = \{ \rho^{\circ l} \mid \rho \in P_H\}$
- $Vis_H = \bigcup_{M \in F_H} Vis(M) \quad Lin_H = \bigcup_{M \in F_H} Lin(M)$

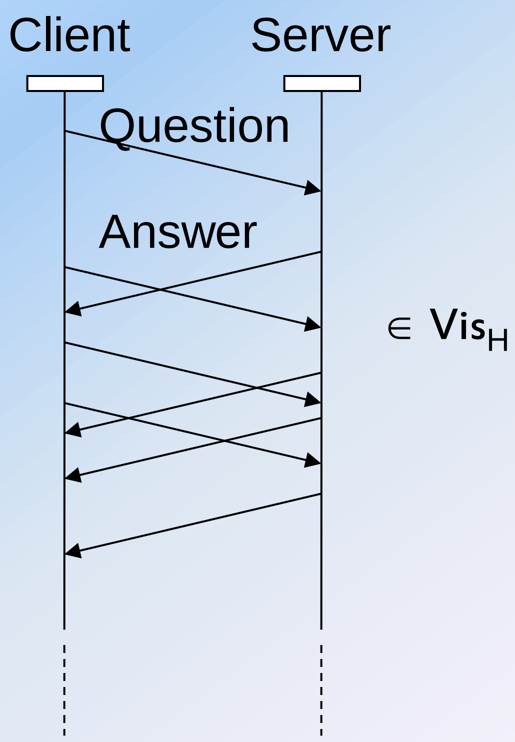
Example

I_{client}= {(*Client!Server(Question), Client?Server(Answer)*)}

I_{Server} = $\varnothing$



A CaHMSC H



Results on Causal HMSCs

- ***Regular*** *CaHMSCs*
  - *Automaton A_H that recognizes Lin_H (exponential size)*
  - *$\cap$, $\subseteq$, =, regular Model checking decidable*

- ***Globally cooperative*** *CaHMSCs*
  - *H CaHMSC and H' with **same** trace alphabet*

    *Build CaHMSCs over **atoms** of H and H'*

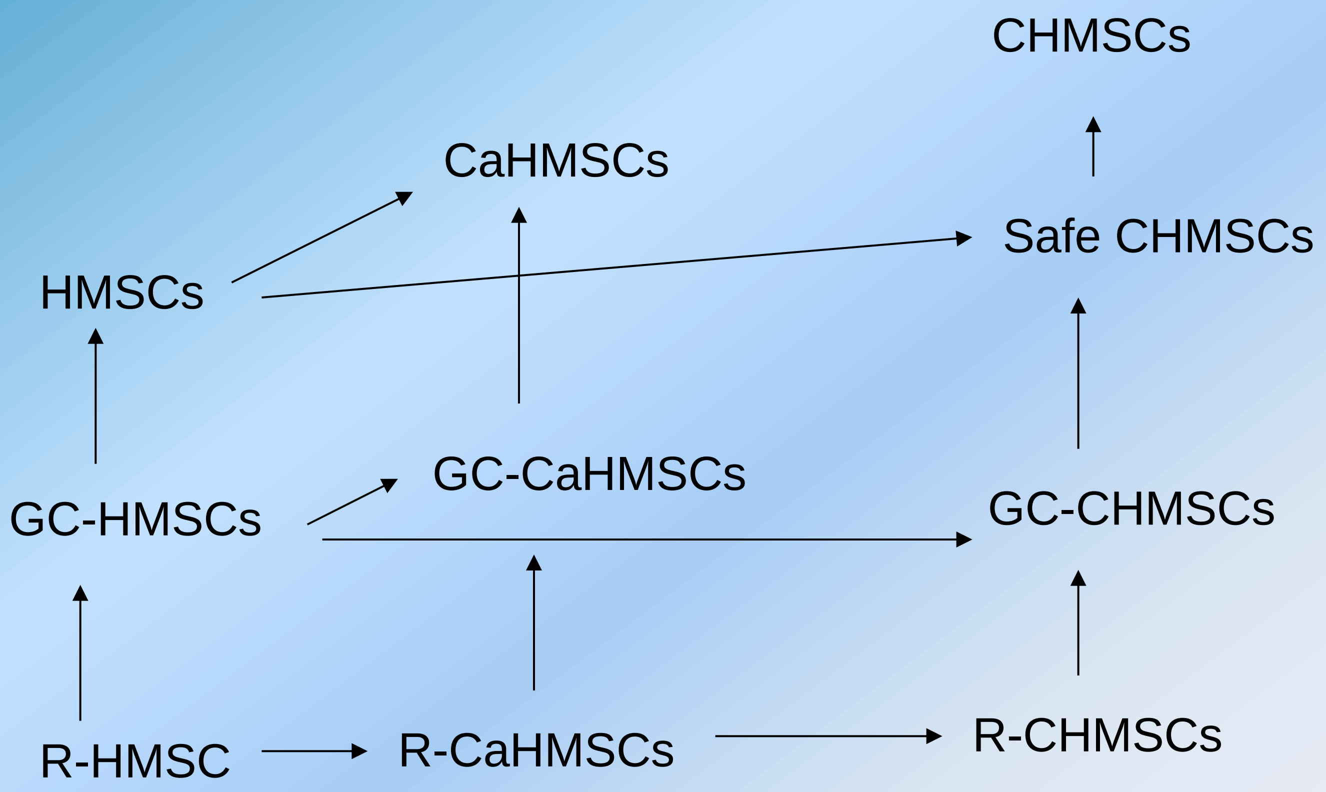
    $F_H \cap F_{H'} = \varnothing$? *PSPACE- complete*

    $F_H \subseteq F_{H'}$ *EXPSPACE- complete*

- *For CaHMSCs H and H' with **distinct** independence relation*

  $F_H \cap F_{H'} = \varnothing$? *undecidable*

Comparison of MSC languages



: Inclusion of MSC Languages

Dynamic MSC grammars

[BH10a]

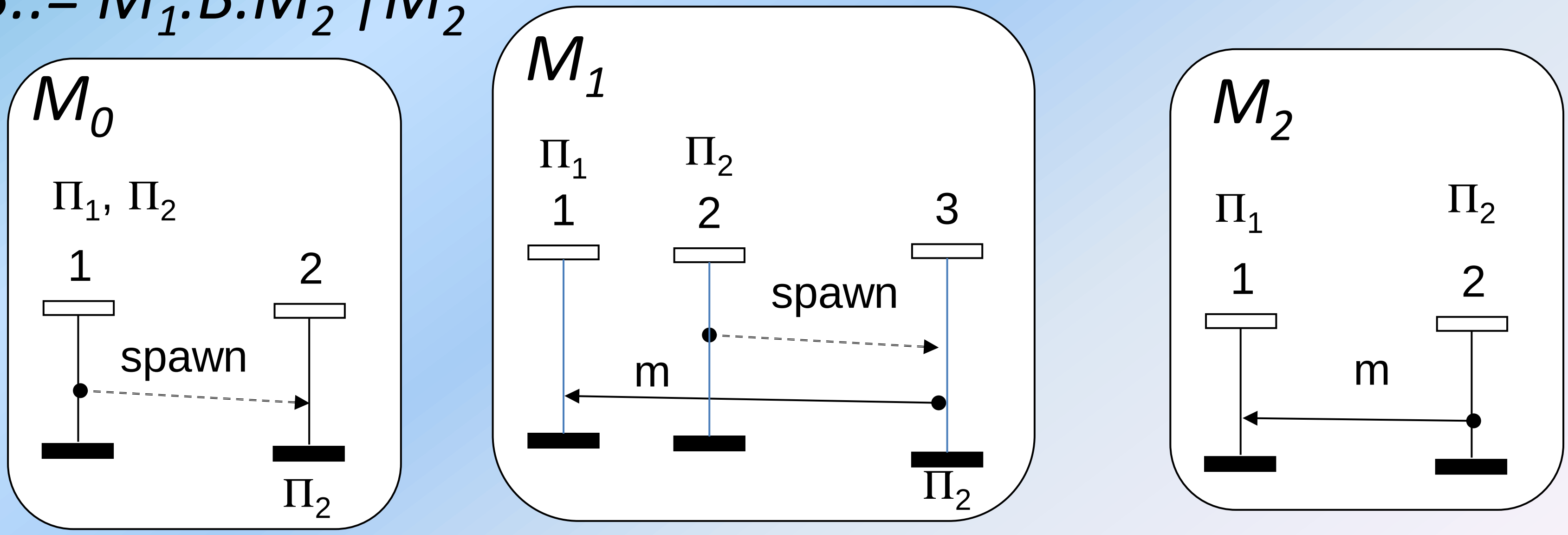
A context free grammar with MSCs as terminals

Axiom ::= M_0. *A*

A ::= M_1.*B* | M_2

B::= M_1.*B*.M_2 |M_2

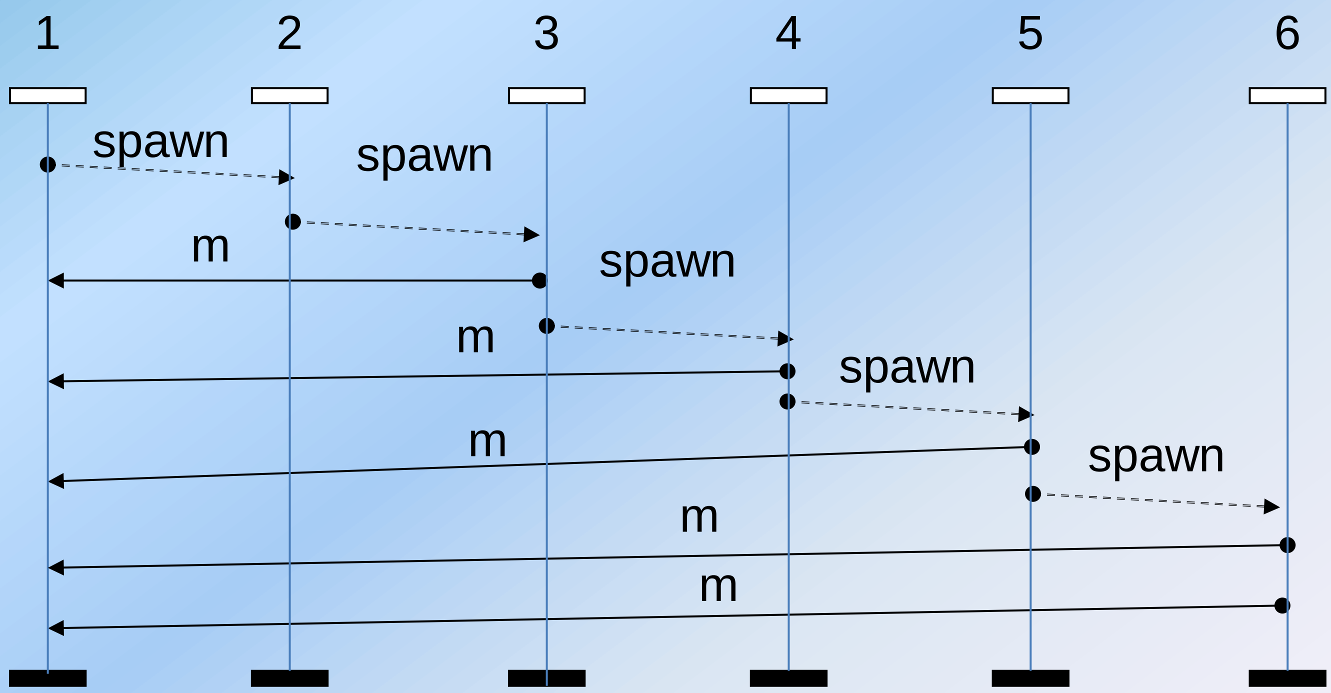
Variant of Dynamic MSCs [Leucker 02]



Dynamic MSC grammars

Generate MSC languages :

- With *dynamic* creation,
- *Over arbitrary* sets of processes

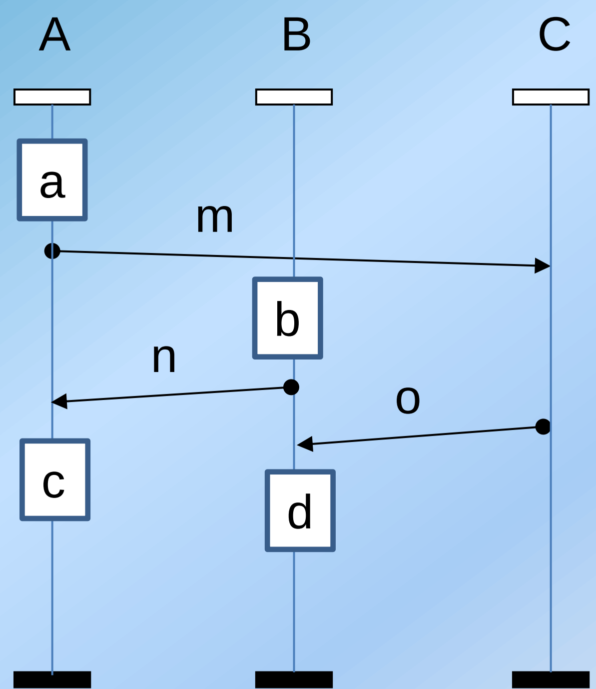


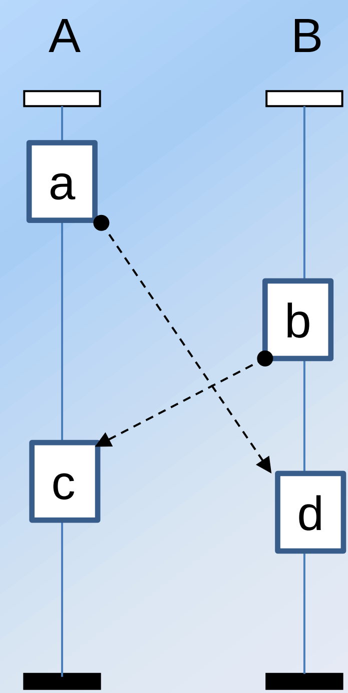
Outline

- Scenario automata
- Extensions
- Operators
- Verification & partial order logics
- Application
- Conclusion

Projection [GHM03]

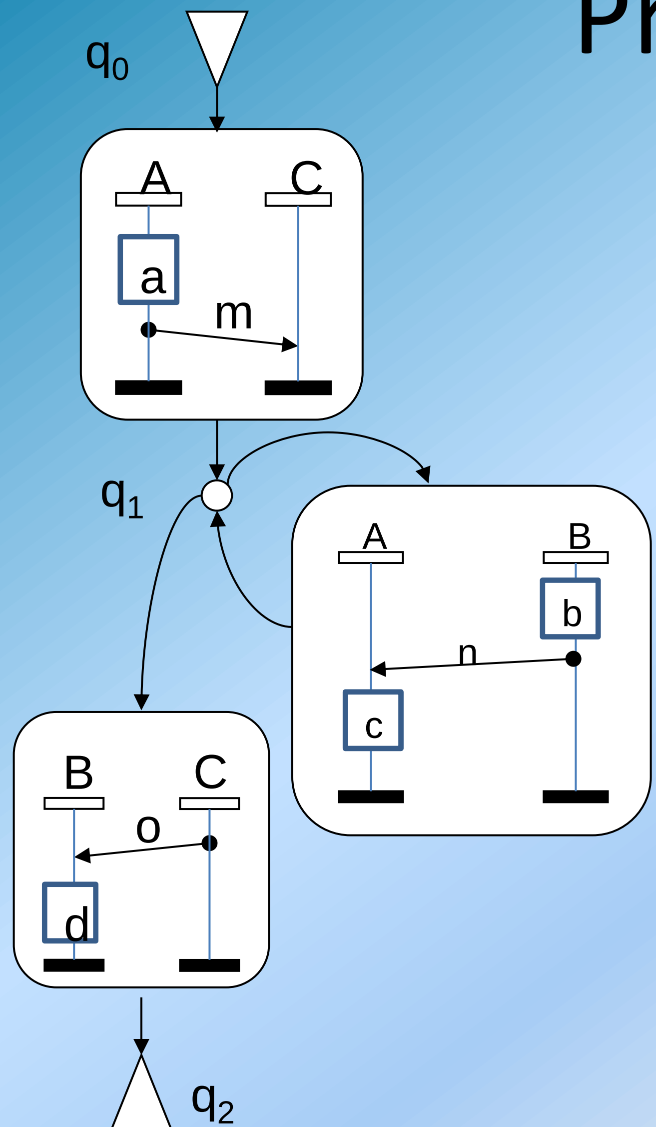
Project MSC languages on a subset of events



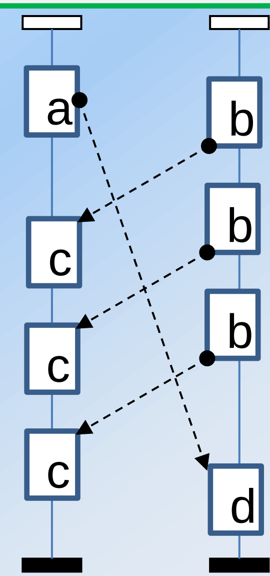


M X={a,b,c,d} $\Pi_X(M)$

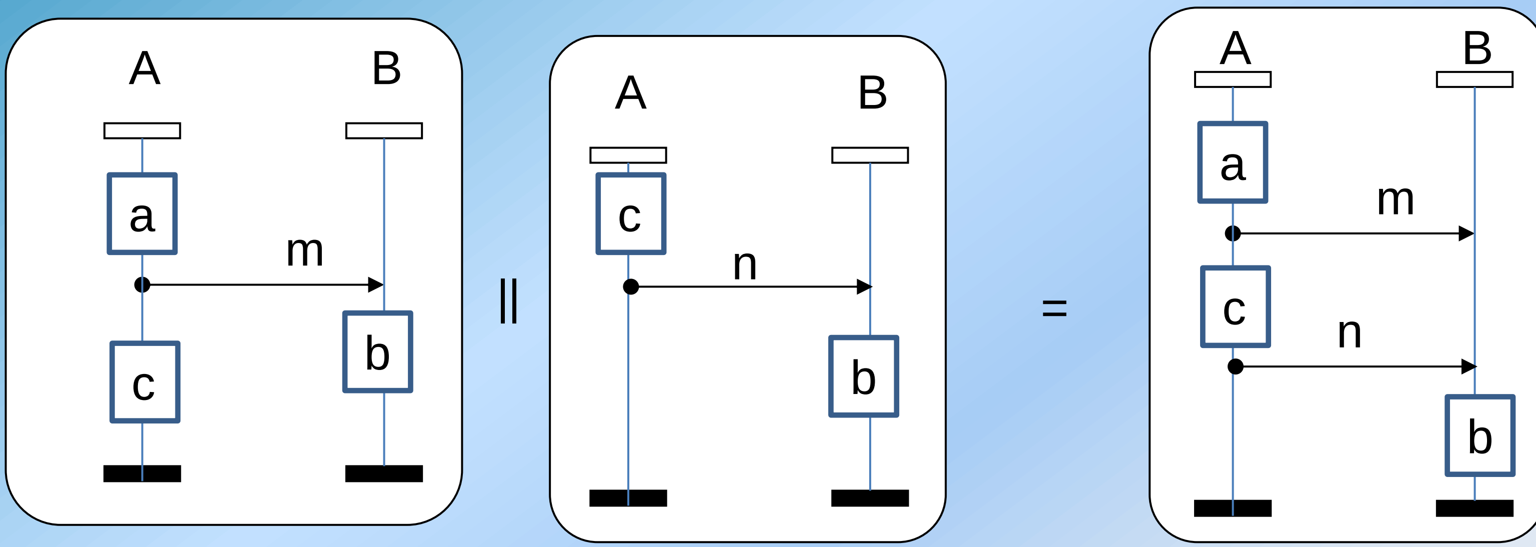
Projection



- HMSC projections and safe CHMSCs have the same expressive power
  $\Pi(F_H) \cap F_{H'} = \emptyset$ decidable (if H' G.C.)
- One can decide if a projection is equivalent to an HMSC



Parallel composition [DGH08]



$$M_1 \| M_2 = \begin{cases} \{ M \mid lin(M) \text{ shuffle of } lin(M_1), lin(M_2) \} \\ \emptyset \quad \text{if } M_1, M_2 \text{ disagree on order of common evts} \end{cases}$$

$$F_{H_1 \| H_2} = \bigcup_{\substack{M_1 \in F_{H_1} \\ M_2 \in F_{H_2}}} M_1 \| M_2$$

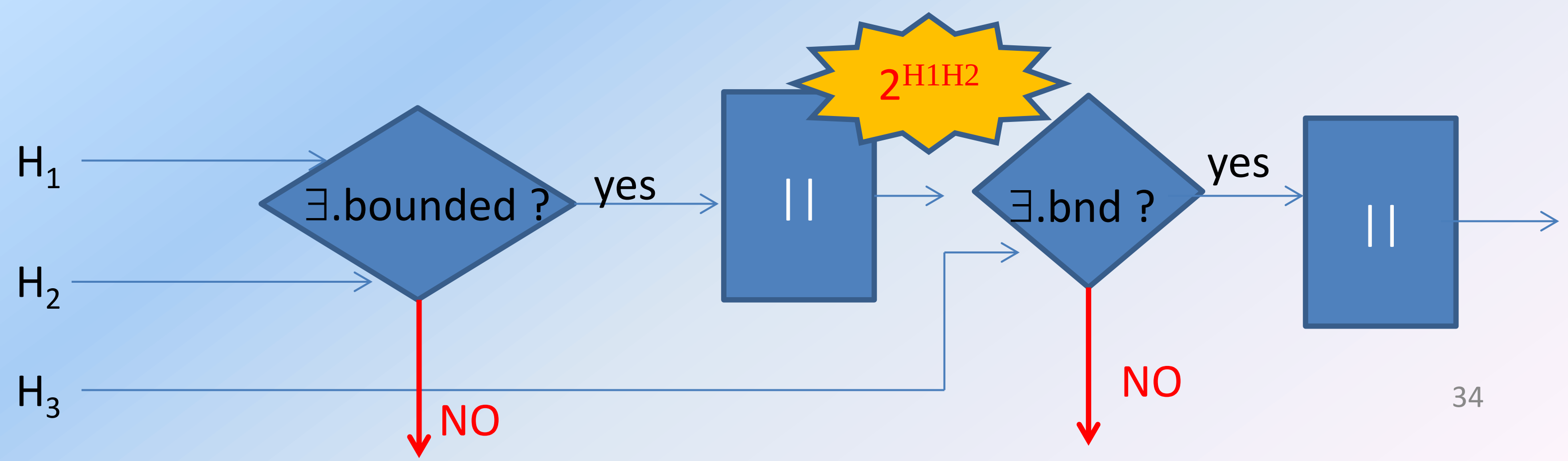
<u>Problem :</u>

$F_{H_1 \| H_2} = \emptyset ?$ undecidable !

Parallel Composition

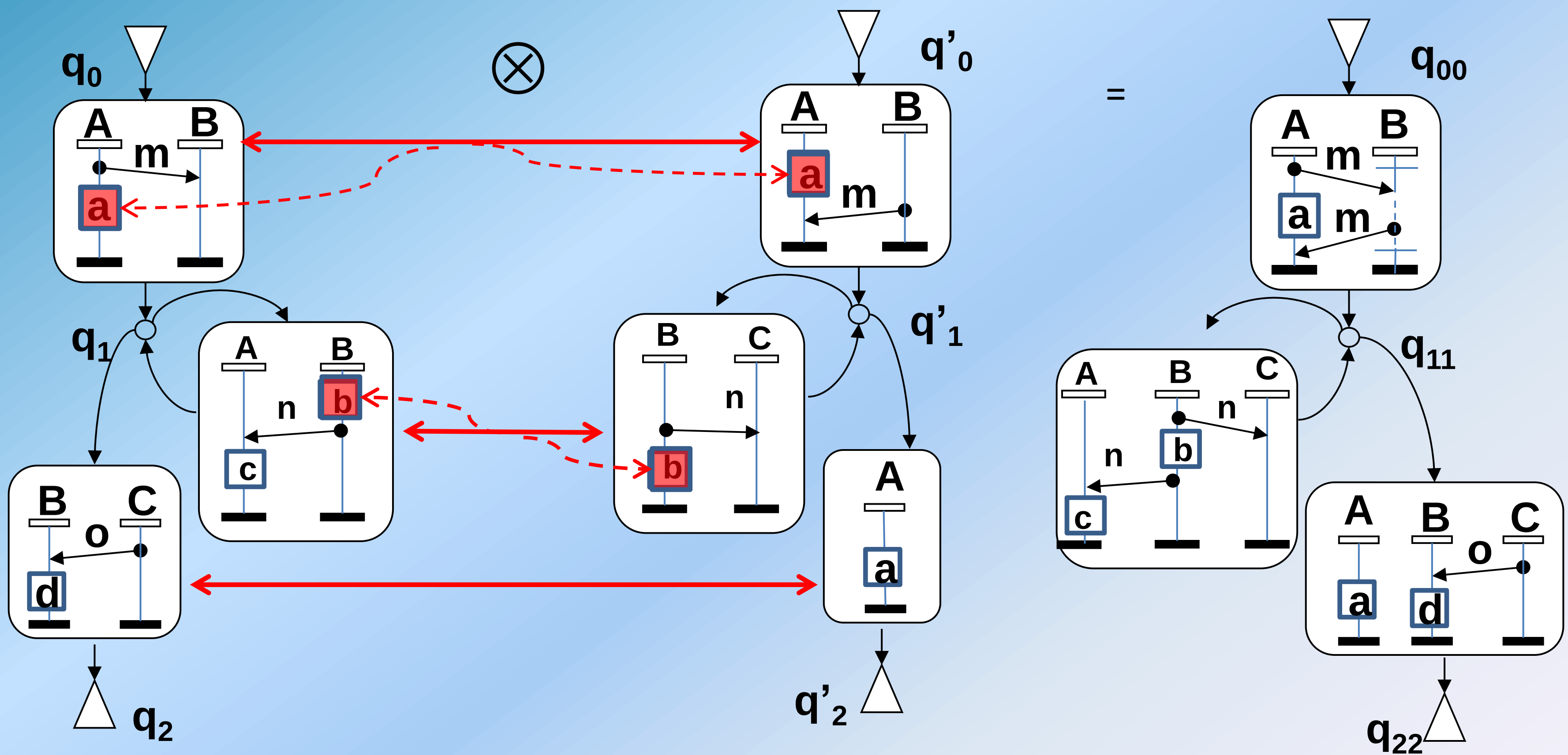
Enforce common events on a single process

- Emptiness for HMSC products decidable (PSPACE)
- $H_1 \| H_2$ existentially bounded decidable
- If H_1, H_2 G. C. and $H_1 \| H_2$ is ∃.bounded, then one can compute an ∃ bounded cHMSC representing $H_1 \| H_2$ (but of exponential size)



Fibered product [CHK04,HKJ06]

Synchronize : MSCs and events in pairs of MSCs



Very syntactic… but more controllable than || : $F_{H_1 \otimes H_2} = \varnothing$? *decidable*

Operators : parallel composition

Conclusion

- HMSC projections can be of practical use

  For verification purpose, to emphasize some causalities,…

- No satisfactory solution for parallel composition: undecidable emptiness, effective product for pairs of GC HMSCs but not beyond, ….
- Interesting subclasses of C/HMSCs are not preserved by Π, $||$, $\otimes$…

Outline

- Scenario automata
- Extensions
- Operators
- Verification & partial order logics
- Application
- Conclusion

Verification

Regular Model-checking (LTL, CTL, etc.) applies only to **regular** [H/C/CA]-HMSCs

Undecidable otherwise

HMSC-based verification:

Let H_{spec}, H_{bad} be two C/Ca/HMSCs. If H_{bad} is **globally cooperative**, then use $H_{spec} \cap H_{bad} = \emptyset$? **(Safety)**

Let H_{spec}, H_{good} be two C/Ca/HMSCs. If H_{good} is **globally cooperative**, then $H_{spec} \subseteq H_{good}$? **(Refinement)**

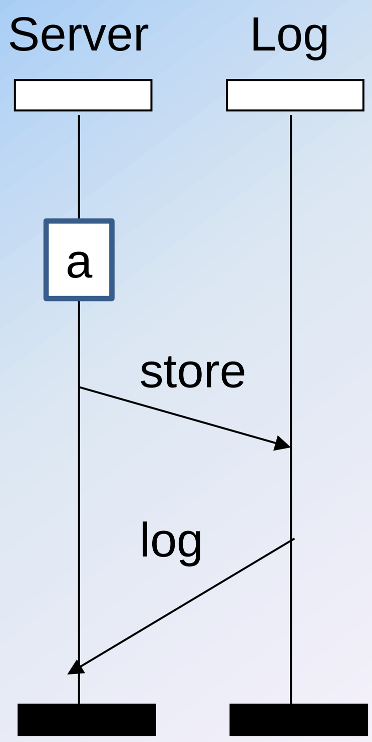
Problem : H_{spec}, H_{bad} , H_{good} need to be defined at the same abstraction level

H Glob.Coop. $\not\Leftrightarrow$ $\Pi(H)$ Glob.Coop.

Verification

MSCs are non- interleaved models :

reason on non-interleaved representation of behaviors
with LOCAL logics (MSO,PDL,TLC^{-},...)



MSO for MSCs

[Madhusudan01]

$$\varphi ::= lab_a(x) \mid msg((x,p),(y,q)) \mid next(x,y) \mid x \in X \mid p \in P$$

$$\neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \exists x, \varphi \mid \exists X, \varphi \mid \exists p, \varphi \mid \exists P, \varphi$$

Verification

Example : MSCs are FIFO.

$$\varphi := \neg(\exists x, x', y, y', p, q, msg((x, p)(y, q)) \wedge msg((x, p)(y, q)) \wedge next(x, x') \wedge next(y, y'))$$

MSO for MSCs is decidable for

- HMSCS [Madhusudan01]
- Safe CHMSCs [MadhusudanMeenakshi01]
- causal HMSCs
- Dynamic HMSCs, Dynamic MSC grammars [Leucker02] [BHH10b]

Main principle : build a (tree) automaton that recognizes models for ˥φ, intersect with the original model. Automata guess an interpretation for variables, synthesizes facts to recognize sequences of MSCs (parse trees) satisfying ˥φ.

Verification

MSO decidability is not so surprising:

all orders produced by these models can be seen as productions of a context free graph grammar.

Can we go further? : specify with logics and drop the automata/grammars/…

[Gastin03]

It is undecidable wheter, given an MSO formula φ, there exists an MSC satisfying φ.

Specifying with logical statements only is not possible. [YHG08]

Outline

- Scenario automata
- Extensions
- Operators
- Verification & partial order logics
- Application : Diagnosis
- Conclusion

Applications : Diagnosis

[GGH06,GGHM13]

Objective :

provide information to supervisors of a distributed system starting from:

- a partial observation (log) : O
- a model of the system : M

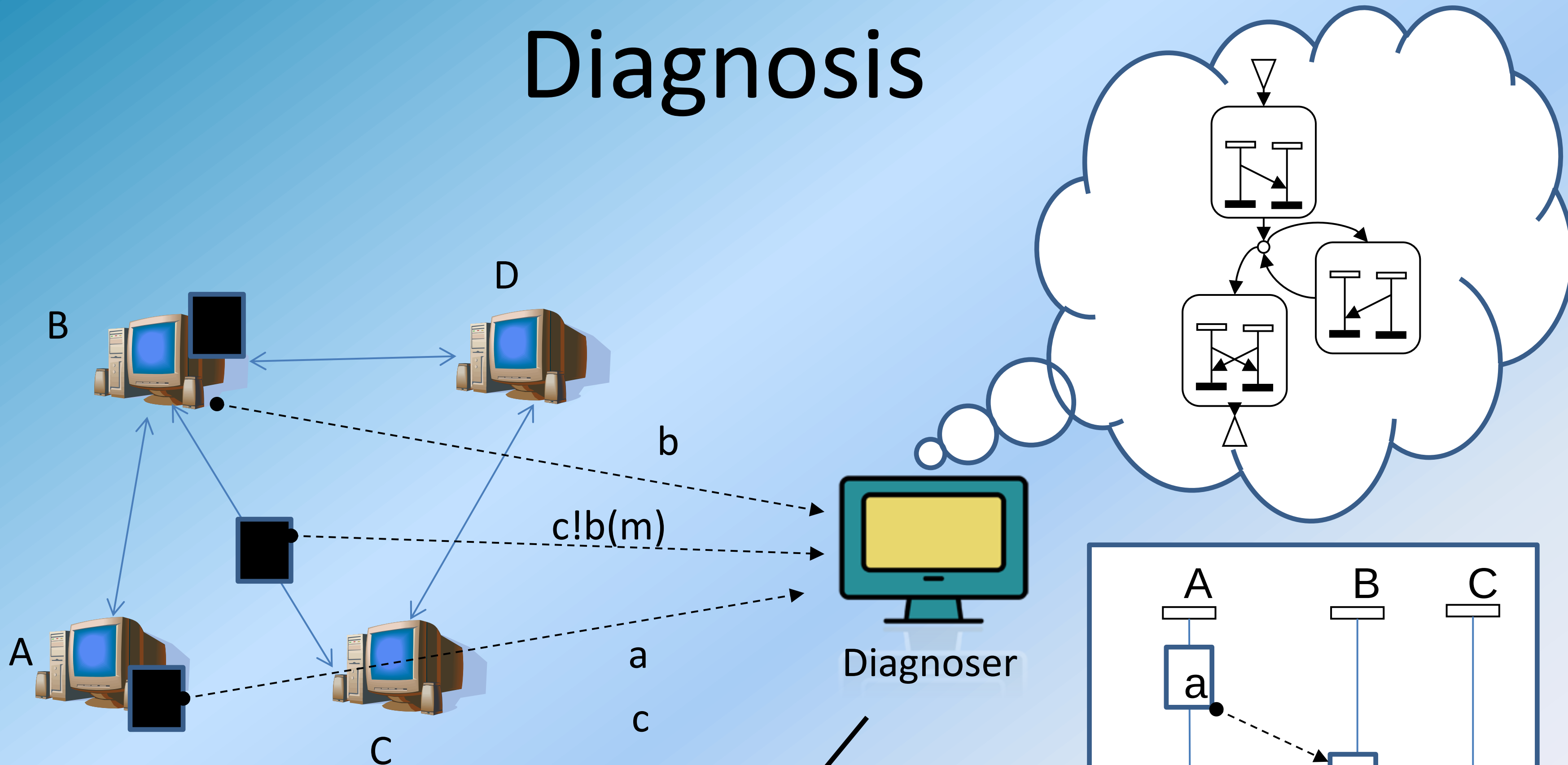
Questions :

- Is there a run of M that embeds O ? **(existence)**
- list all runs that embed O ? **(Diagnosis)**
- is there a faulty run that embeds O ? **(fault detect°)**

Solutions for : Automata, Petri Nets, …

Build a product or an unfolding : M x O

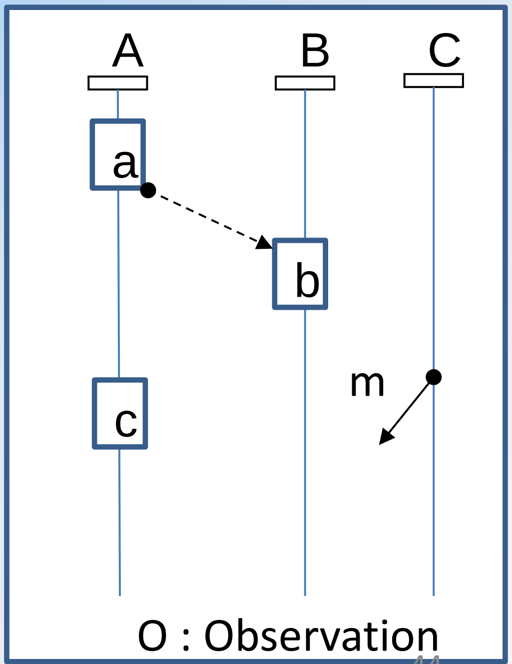
Diagnosis



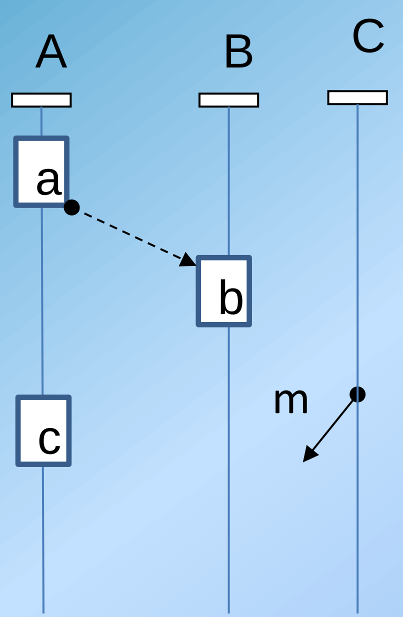
Probes

Diagnosis : H'= H x O

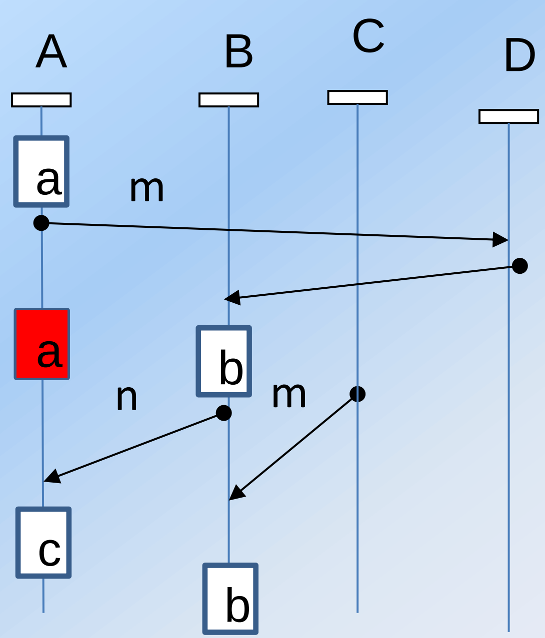
Every M in $F_{H'}$ embeds O



Embedding



O : Observation



Explanation for O as a MSC

Results

<u>Theorem:</u>

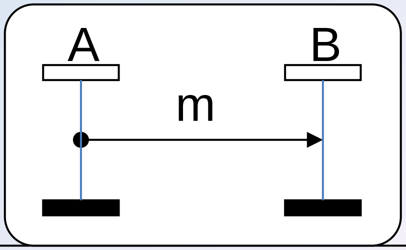
H' = H x O is an HMSC of size at most in $O\left(|H|.|O|^{|P|\times|P_{obs}|}\right)$ s.t.

- every M in F_H that embeds O is also in $F_{H'}$
- every M in $F_{H'}$ embeds O

- Diagnosis/existence are decidable for HMSCs
- Nice associative properties : $(HxO_1) \otimes (HxO_2) = H \text{ x } (O_1 \,||\, O_2)$

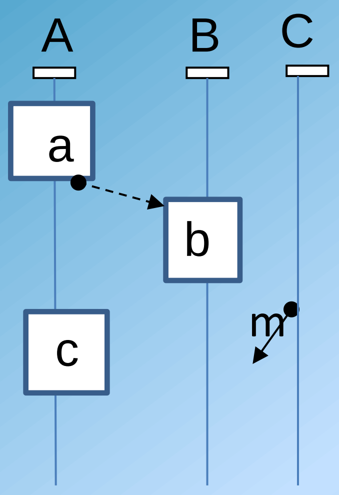
<u>Note</u> : undecidable in general for CHMSCs

Take G a cHMSC: is there a run that embeds O
= *The simple message problem*

O = 

Application:Diagnosis

Diagnosis as an MSO property



$$\varphi_O := \exists x, y, z, t, lab(x,y,z,t) \wedge caus(x,y,z,t) \wedge closed(x,y,z,t)$$

$$lab(x,y,z,t) := lab_a(x) \wedge lab_b(y) \wedge lab_c(z) \wedge lab_{!m}(t)$$

$$caus(x,y,z,t) := x \leq y \wedge x \leq z$$

$$closed(x,y,z,t) := \forall x', x' \prec x \vee x' \prec y \vee x \prec x' \prec z \vee x' \prec t \Rightarrow lab_{\Sigma obs}(x')$$

M embeds O $\Leftrightarrow$ M $\models$ φ_O

Diagnosis / existence are decidable for CaHMSCs, dynamic HMSCs,…

Outline

- Scenario automata
- Extensions
- Operators
- Verification & partial order logics
- Application
- Conclusion

Over the last 10 years

- Several extensions : Causal HMSCs, dynamic MSC Grammars, Extended coregions, ….

- Decidable classes remain incomplete models.
- Composition w/o automaton/grammar structure highly undecidable
- Modularity (//) leads to undecidability too

- MSO decidability / diagnosis easily achieved

- Scenarios well adapted to Diagnosis.

The Scenario Algorithm

Repeat

[C/Ca/Dyn/…] HMSCs are incomplete

Propose an extension !

Pb *X* is undecidable for the extension

Find a subclass to solve the problem

Until ???

??? =
(provocation)

- ❑ the model is complete enough to be a decent modeling tool
- ❑ your models are so ugly that nobody wants to use them
- ❑ no one wants to read a paper on MSCs anymore
- ❑ you get bored
- ❑ ….

Future plans

Scenario specific issues :

- **Generalize** decidability results for scenarios seen as graph grammars

- **Implementation**: for HMSCs, Dynamic MSC grammars
- **Causal HMSCs:** finite generation, equivalence with HMSCs
- **Diagnosis:** Online diagnosis, alternatives to MSO
- **Security:** covert flows detection with probabilities, information theory, etc.

Future plans

- **Time & Robustness :**

  Check if timed requirements make sense w.r.t. implementation assumptions (architecture,…)

- **Abstraction :**

  obtain decidable models using s abstractions of real systems

From Telecoms to services:

many assumptions on scenarios due to IP-like communications (FIFO, etc)

In **Services** : no FIFO, open and dynamic world, sessions,data…

new models, new problems, new solutions

Closer to real world situations

Acknowledgements

- The Distribcom/Vertecs/S4 (now SUMO) colleagues
- Students and Postdocs
  - A. Degorre
  - T. Gazagnaire, J.Klein, T.Ziadi, R. Abdallah,…
  - S.Yang, S.Akshay, B. Masson
- All members of the CASDS, DST, and now DISTOL associated teams, INRIA for funding them

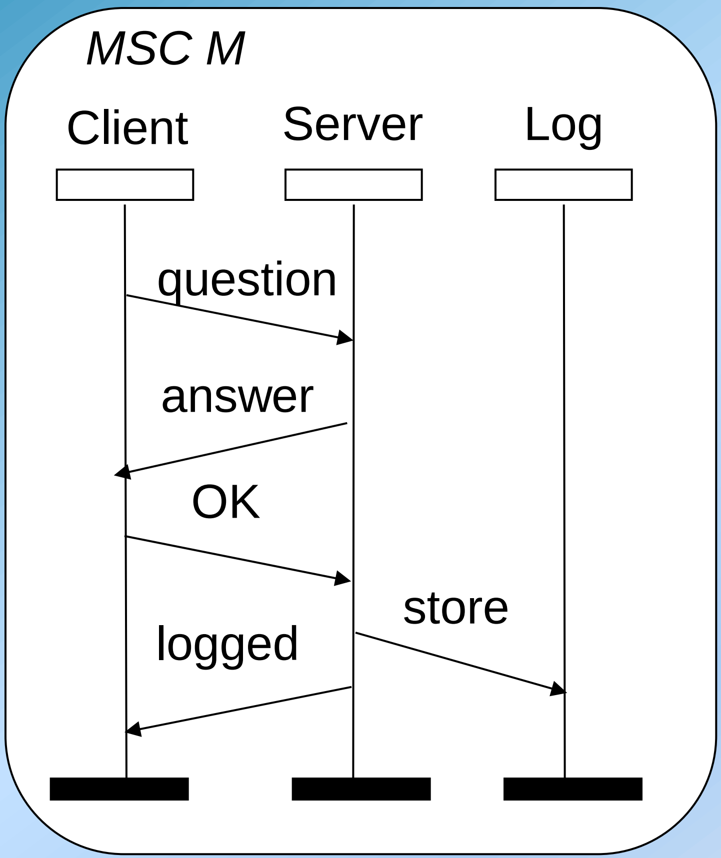
- courir@inria.fr, courir@irisa.fr

And of course….Philippe

Questions ?

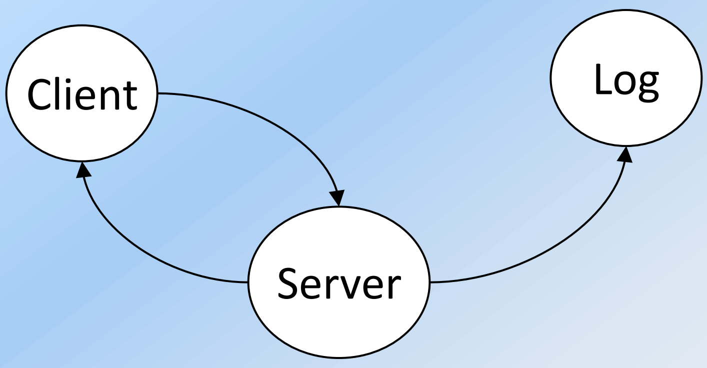
Ravitaillement en salle **Sein** !

Communication graph



An example MSC M

An useful abstraction of the contents of MSCs



Communication graph CG(M)

Regular HMSCs

[Alur et al 99] [Muscholl et al 99]

<u>*Definition:*</u> H is regular iff

$\forall \rho = q_1 \xrightarrow{M_1} q_2 \xrightarrow{M_2} \ldots \xrightarrow{M_k} q_1$ cycle of H,

$CG(\rho^\circ)$ is a strongly connected graph

<u>*Theorem:*</u>

H regular $\Rightarrow$ Lin_H regular subset of Σ^*

<u>Consequences:</u>

$Lin_{H1} \cap Lin_{H2} = \emptyset$, $Lin_{H1} \subseteq Lin_{H2}$, $Lin_{H1} = Lin_{H2}$

$R \subseteq Lin_{H1}$, $Lin_{H1} \subseteq R$ decidable

Globally cooperative HMSCs

[Genest et al 02][Morin02]

Definition: H is Globally Cooperative iff

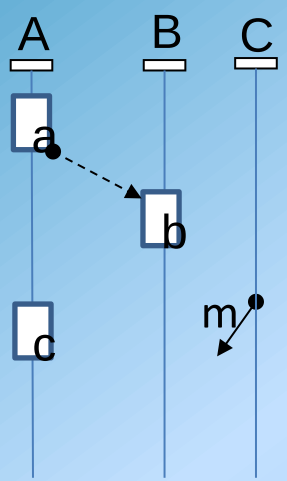
$\forall \rho$, cycle of H, $CG(\rho^{\circ})$ is a connected graph

Theorem: [Genest et al 02]

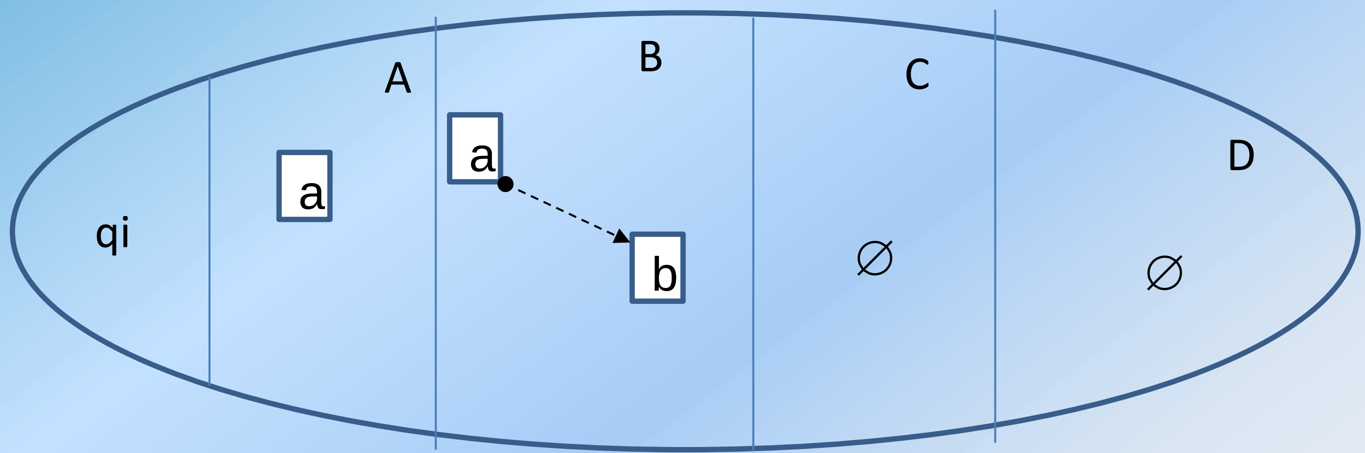
Let H_1 be a HMSC,
H_2 be a globally cooperative HMSC, then

$F_{H_1} \cap F_{H_2} = \emptyset$?,
$F_{H_1} \subseteq F_{H_2}$? Decidable

Product : states



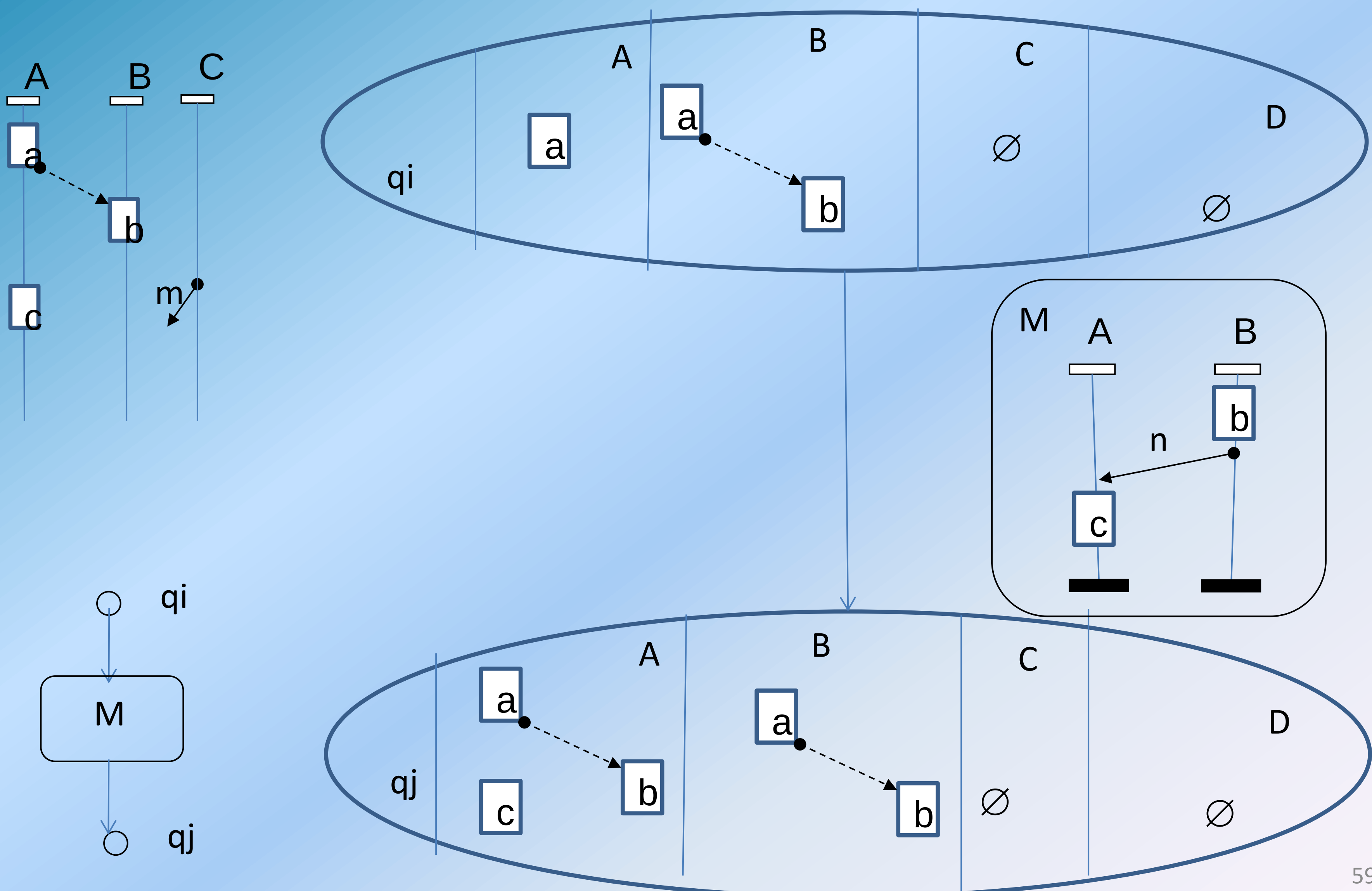
O : Observation



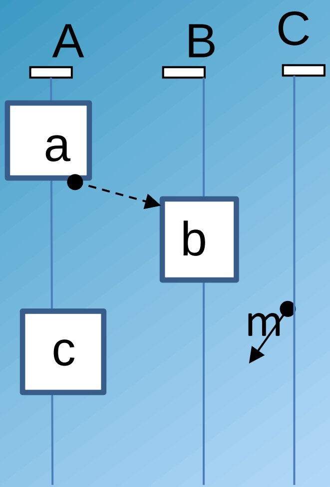
<u>States recall:</u>

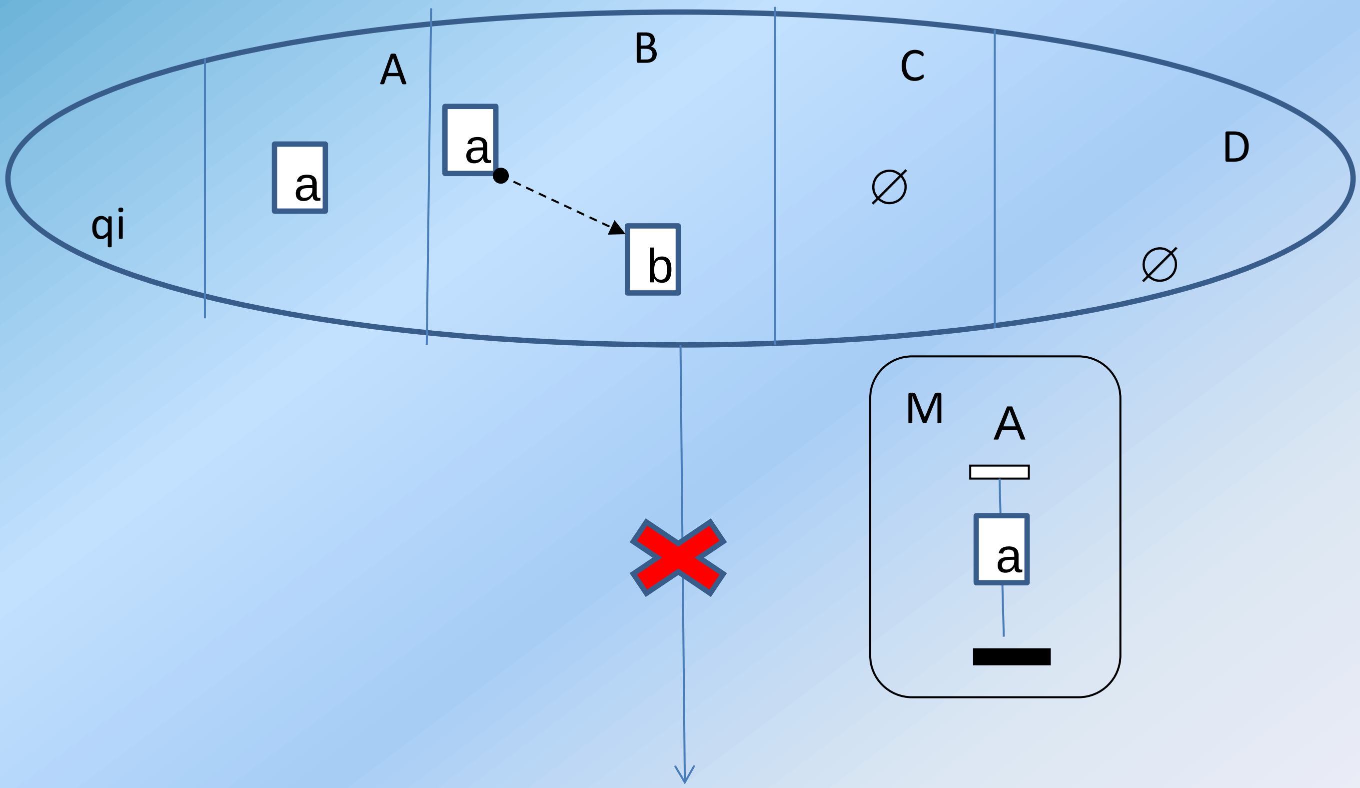
- A node of H
- for every process p, the events of O that have been « seen » by p

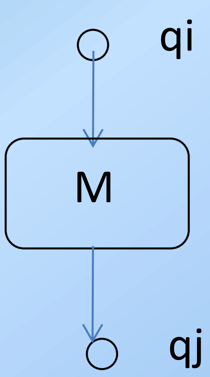
Product : transitions



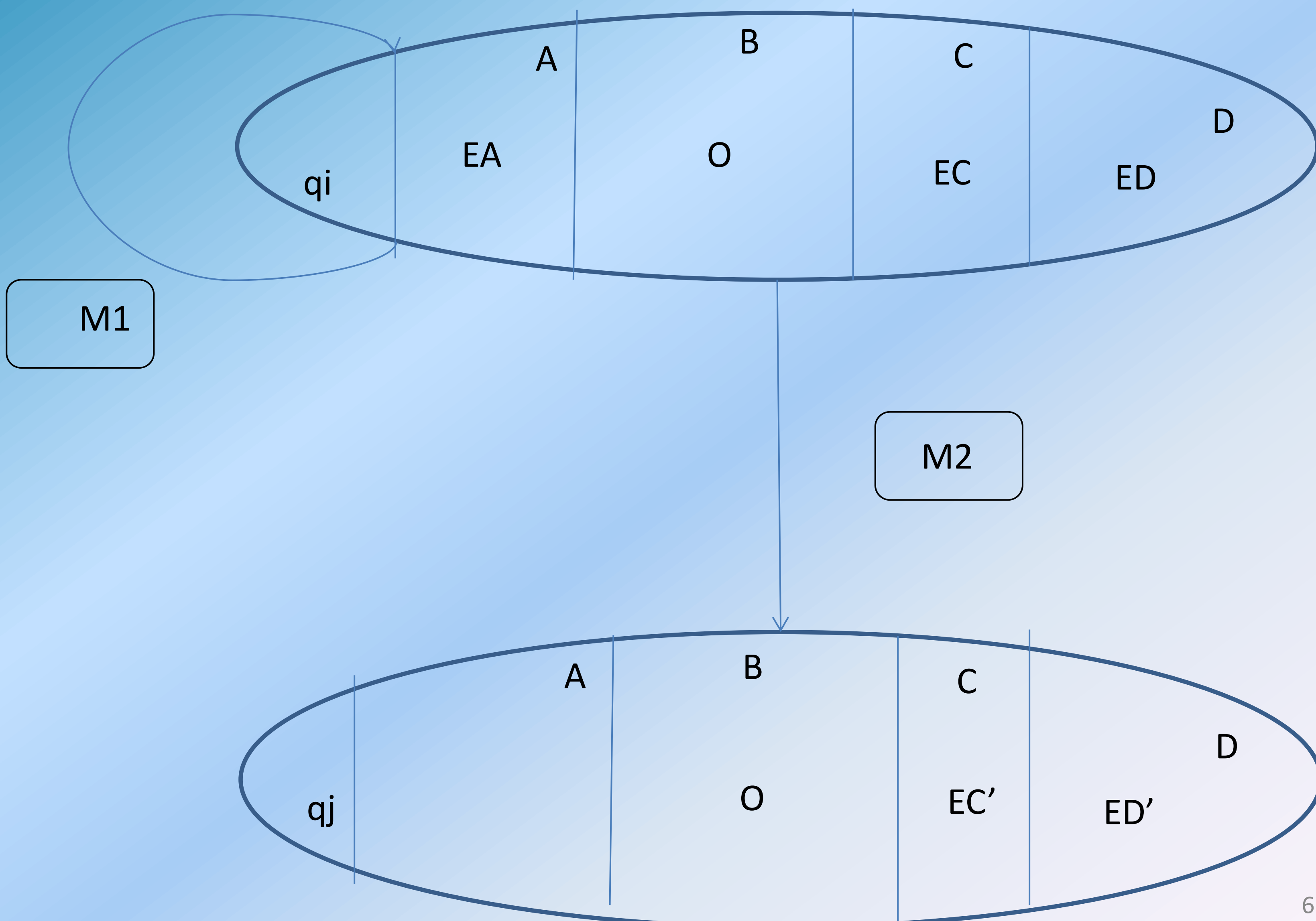
Product : transitions







Product : final nodes



Applications : intrusion detection

<u>Idea</u> : use HMSCs to represent « normal behaviors » of a system

```
Normal (O,H_1,…H_n)
        for (i=0;i<n;i++){
                if (L(H_i × π_{Σ_i}(O)) = ϕ) {
                        return true // raise an alarm
                }
        }
Return false
```

$$\text{if } (L(H_i \times \pi_{\Sigma_i}(O)) = \phi)$$

If no explanation exists for an observation O, raise an alarm.