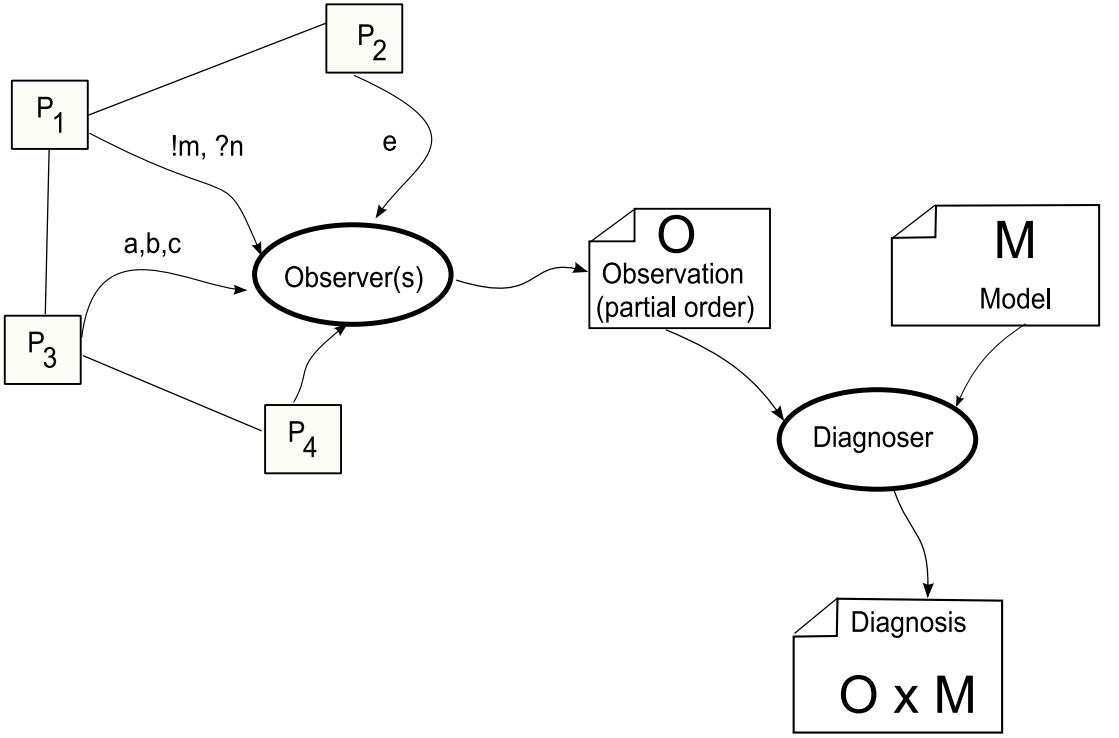
Diagnosis with Dynamic MSCs

(Ongoing work)

Benedikt Bollig[1], Stefan Haar[2], Loïc Hélouët[3]

[1]LSV/CNRS, [2]LSV/INRIA, [3]IRISA/INRIA

Motivations



Observation = code instrumentation,

traffic sniffing & filtering, ...

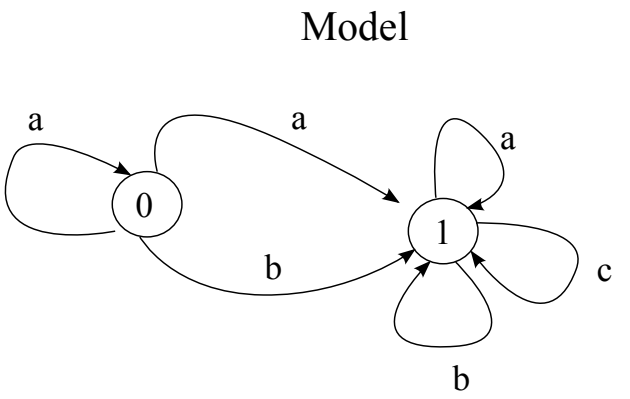
Diagnosis = Fault detection, explanation retreival,...

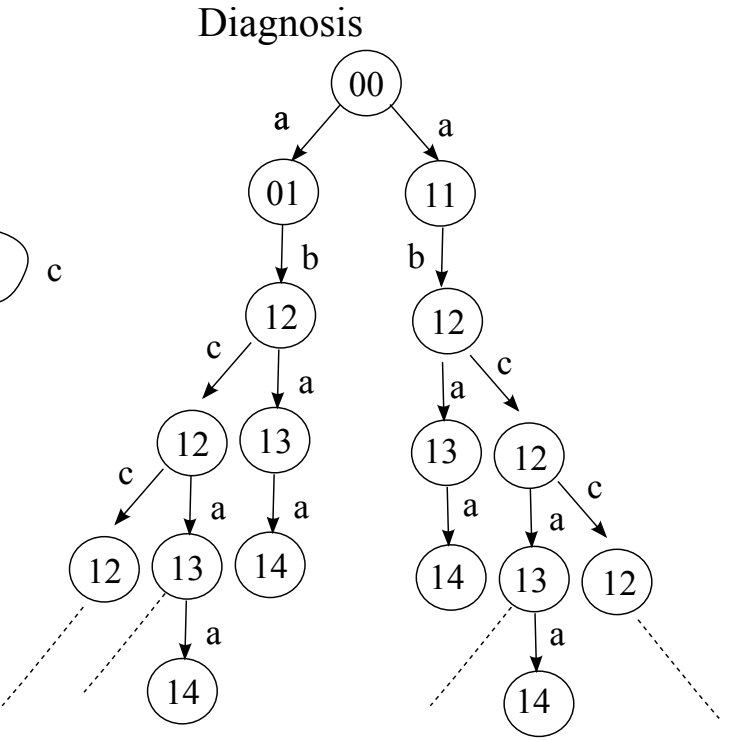
Diagnosis

From a model H and an observation O, find all explanations for O in H.

Observation

0 1 2 3
a . b . a . a





Unfolding: synchronize execution of the observation with all compatible runs of the model (may not terminate)

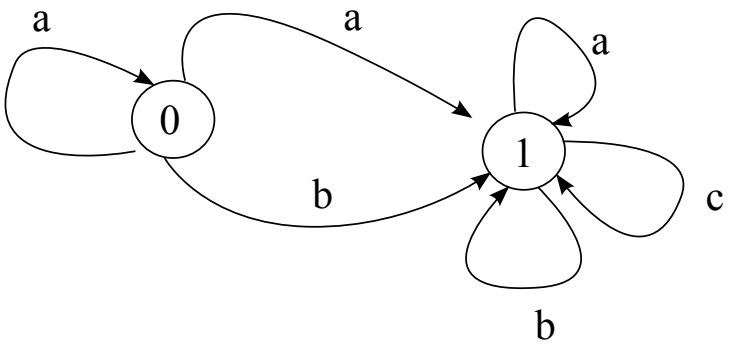
Diagnosis

From a model H and an observation O, find all explanations for O in H.

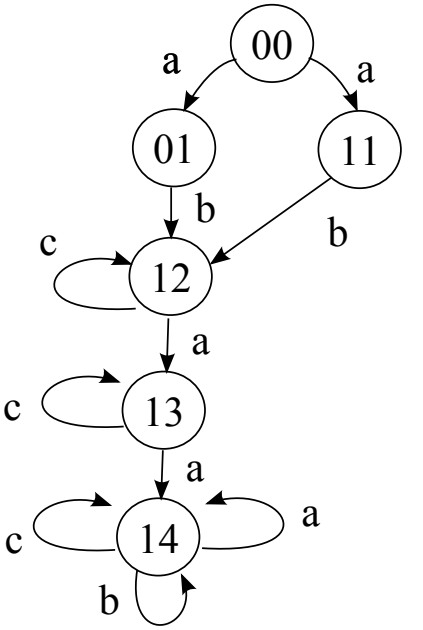
Observation

0 1 2 3
a . b . a . a

Model



Diagnosis



Generator: build a new model in which all behaviors embed the observation (not always feasible)

Motivations

Done for:

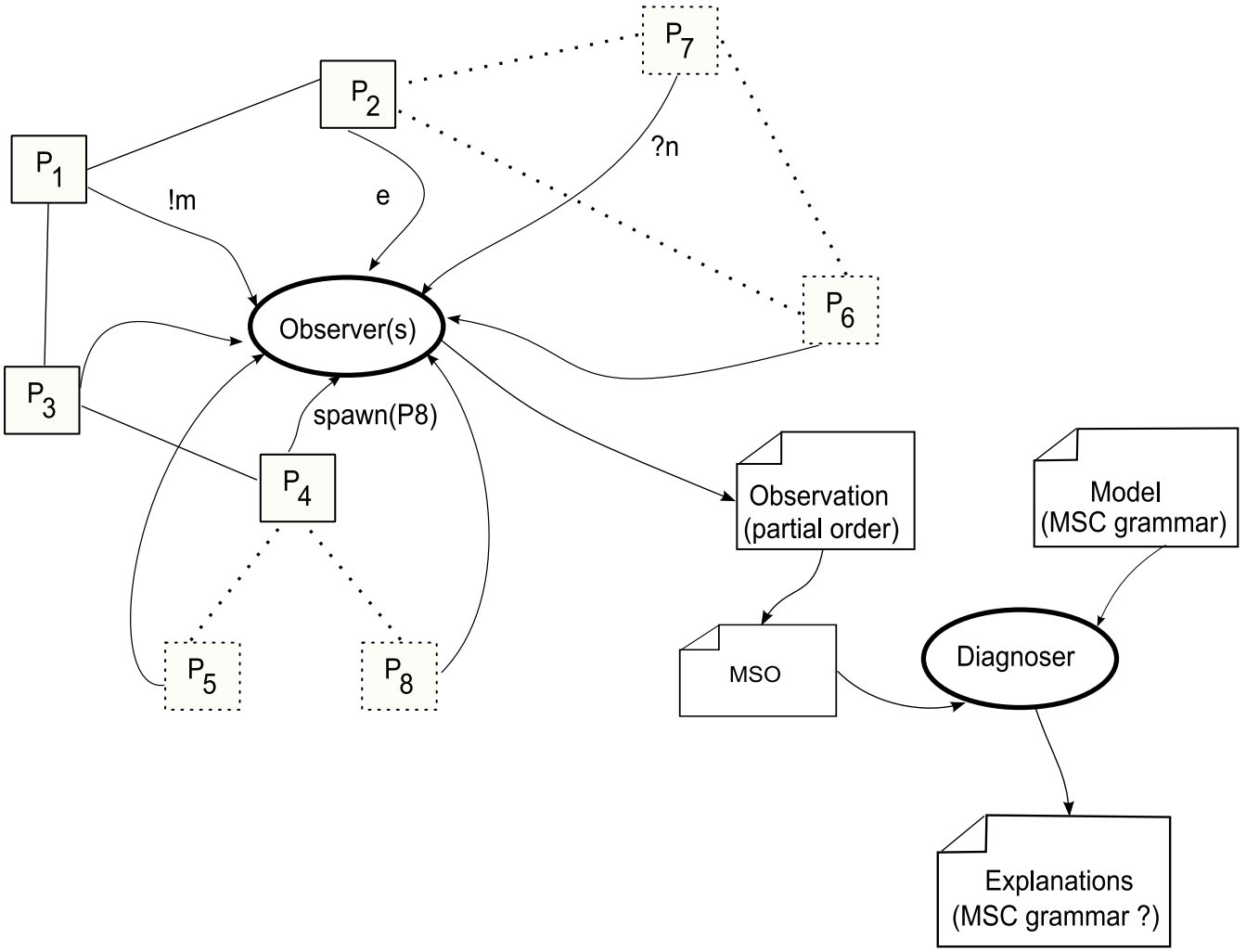
- Automata (static, fault detection) [Sampath & al]
- Petri nets (static archi, unfolding) [Benveniste & al, Chatain & al]
- High-level MSCs (static archi., generator) [Gazagnaire & al]
- Graph grammars (dynamic archi., unfolding) [Baldan & al]

Motivations

Diagnosis for :

- Partial order model (can avoid costly interleavings)
- with dynamic aspects + buffering
- Compute a generator
- Compositionality issues

Motivations



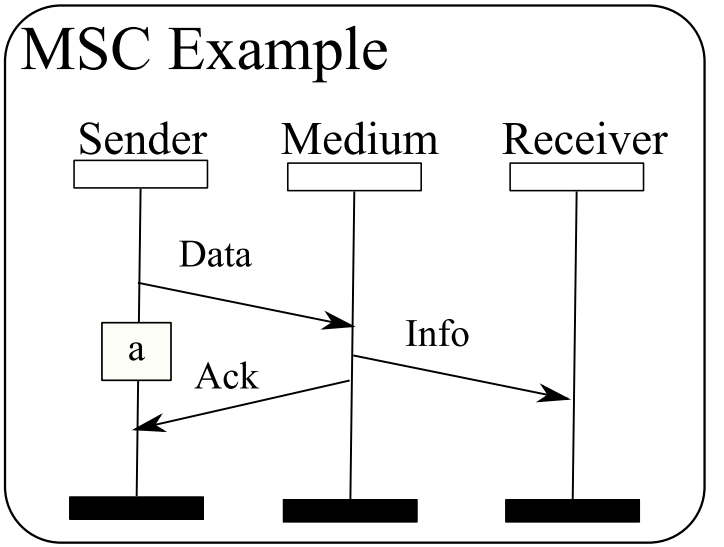
Outline

- Message Sequence Charts, Observations
- MSC Grammars
- MSO for MSCs
- Diagnosis
- Conclusions

Message Sequence Charts

$M = (E, \leq, \alpha, \varphi, \mu) \in \mathbb{M}$:

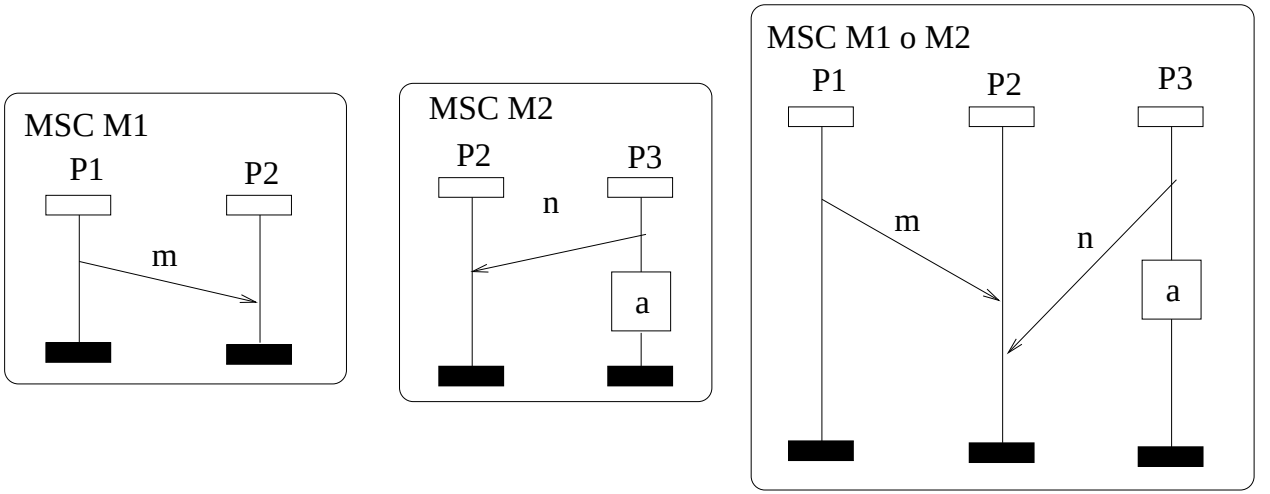
- $\alpha \subseteq E \times \Sigma$: labeling,
- $\varphi \subseteq E \times \mathbb{N}$: locality,
- $\mu \subseteq E \times E$: messages



$x \leq y$ iff

- sequential ordering on a process, or
- causal chain from x to y

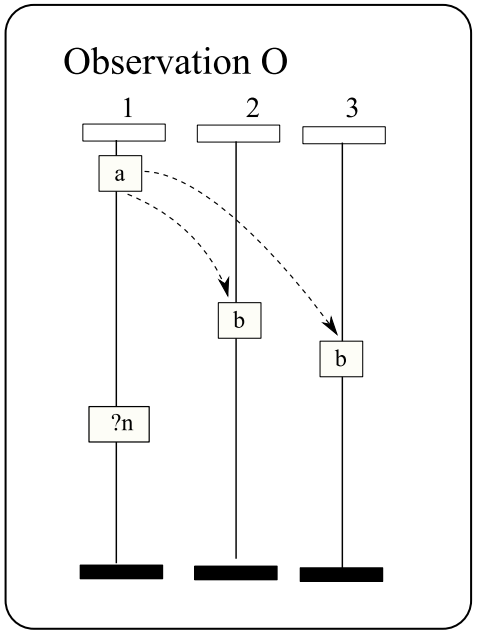
MSC Concatenation



$x \leq_{1 \circ 2} y$ iff

- $x \leq_1 y$, or
- $x \leq_2 y$, or
- $x \leq_1 z$, $z' \leq_2 y$ and z, z' on the same process

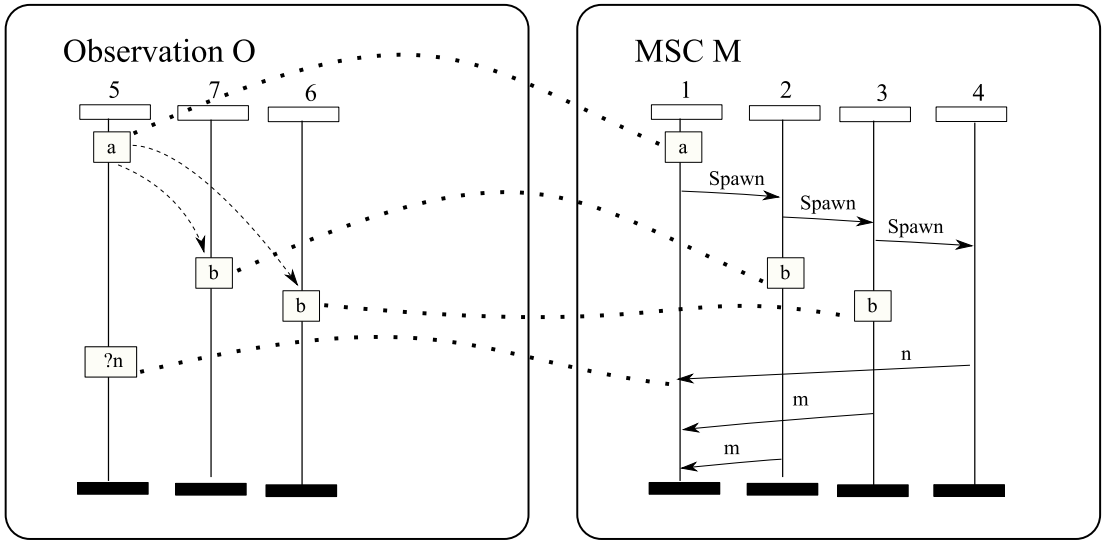
Observations



- Choose an observation alphabet Σ_{obs}
- Every observed process
  - reports occurrences of actions in Σ_{obs}
  - can maintain tags to retreive causality (eg vectorial clocks)
- Observer receives collected events and builds $O = (E_O, \leq_O, \alpha_O, \varphi_O, \mu_O)$

Explanations

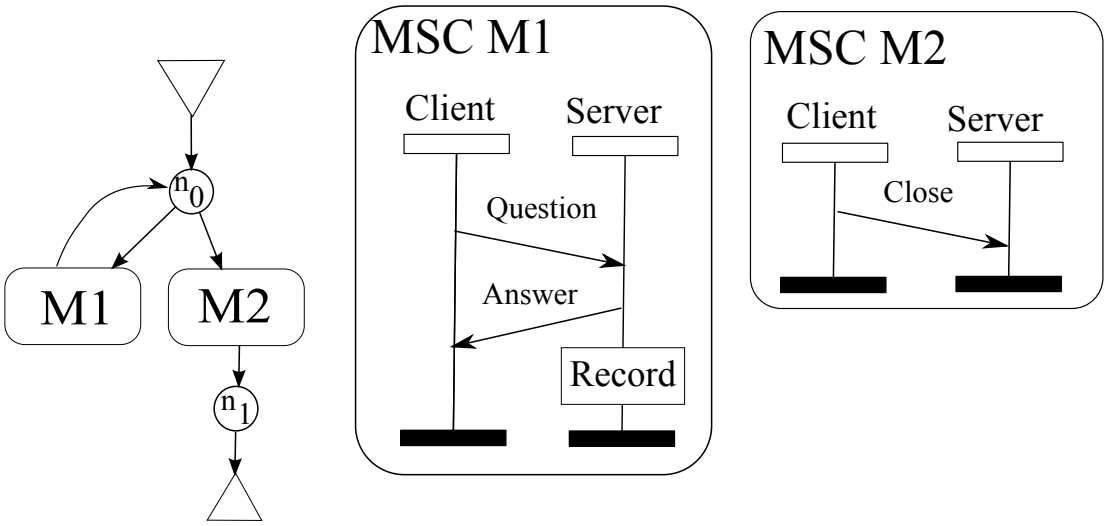
$O \triangleright_{\Sigma_{Obs}} M$



Find an embedding $h : E_O \longrightarrow E_M$ that is compatible:

- with labeling : $\alpha(x) = \alpha(h(x))$
- with ordering : $x \leq_O y \Longrightarrow h(x) \leq_M h(y)$
- with locality of events :
  $\varphi(x) = \varphi(y) \Longleftrightarrow \varphi(h(x)) = \varphi(h(y))$
- ....

High-level MSCs (HMSCs)



- patial order model, infinite, non-regular behaviors
- Diagnosis with generator works [HelouetWodes06]

But : Finite set of processes only

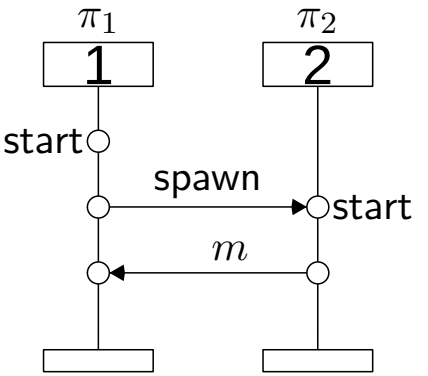
MSC Grammars

The important in MSC grammars is the shape of a scenario, not the identity of processes.

Named MSCs : $\mathfrak{M}$

Rely on process identifiers: $\pi_1, \ldots, \pi_k$.

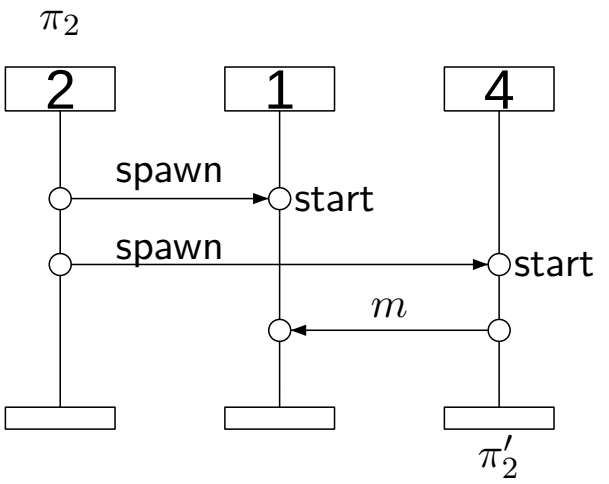
$(M_1, \nu_1) =$



MSC Grammars

Named PMSCs : $\mathcal{M}$

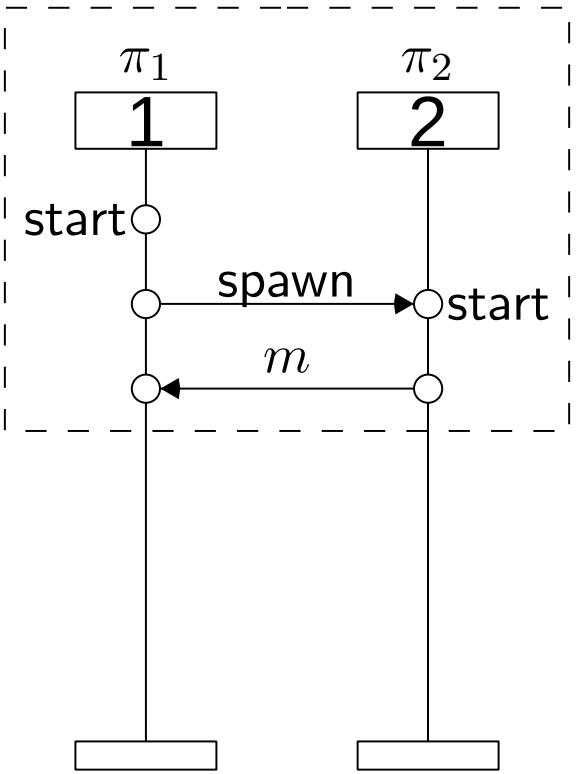
$(M_2, \mu_2) =$



- Process IDs passed from one process to another
- the name of non-identified processes is meaningless
- Glue a behavior on identified processes

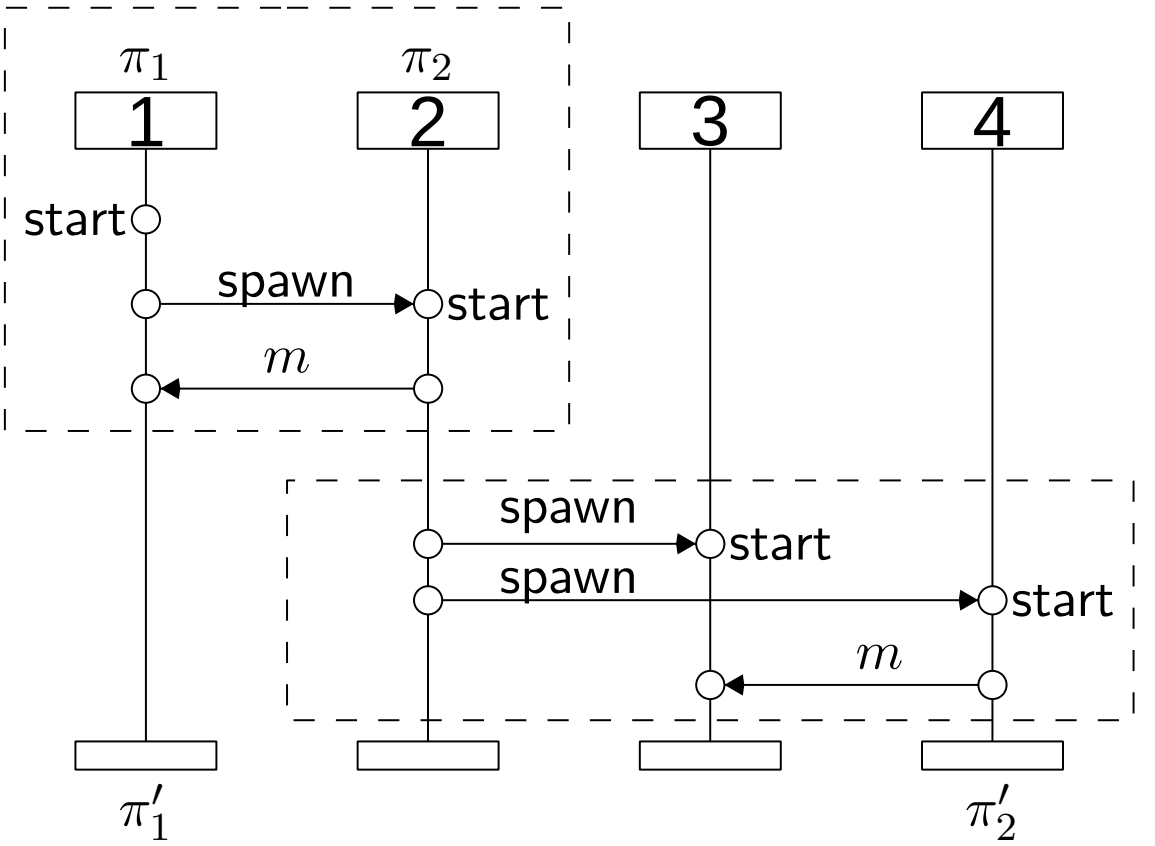
MSC Grammars

Concatenation: $(M_1, \nu_1) \circ (M_2, \mu_2)$



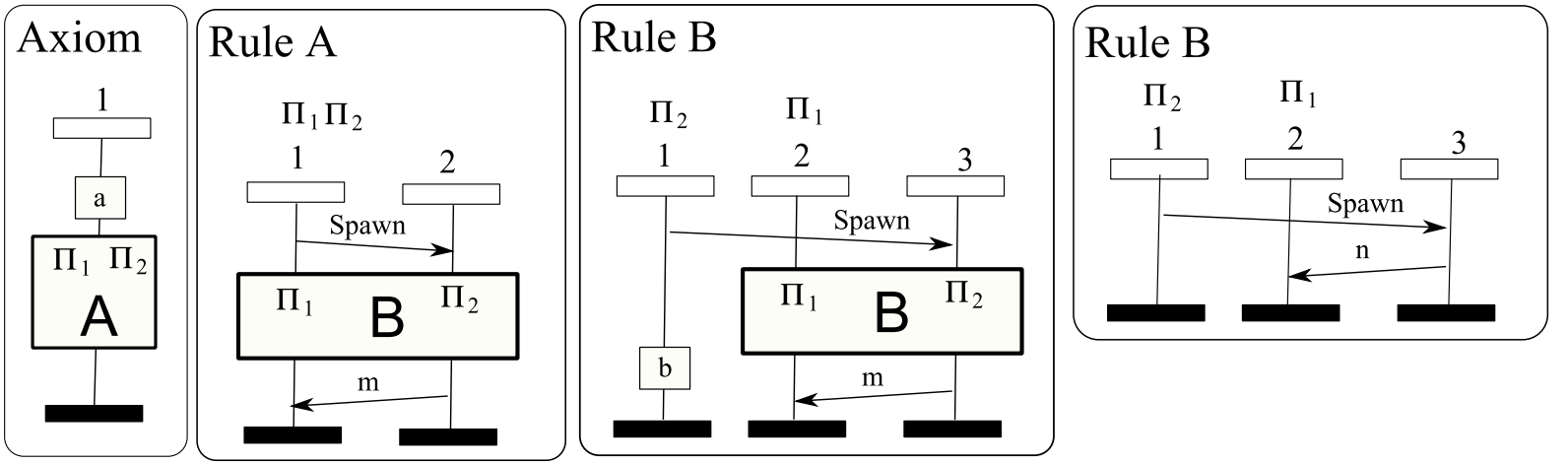
MSC Grammars

Concatenation: $(M_1, \nu_1) \circ (M_2, \mu_2)$



MSC Grammars

In short : context free grammar with named MSCs and PMSC as terminals



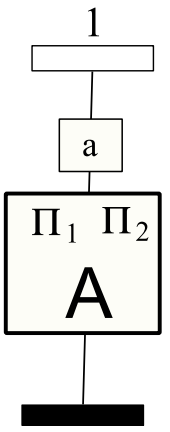
- Subset of Dynamic MSCs [Leucker & al 02]
- A kind of graph grammar...

MSC Grammars

Derivations:

... as usual but writing PMSCs instead of words/hypergraphs

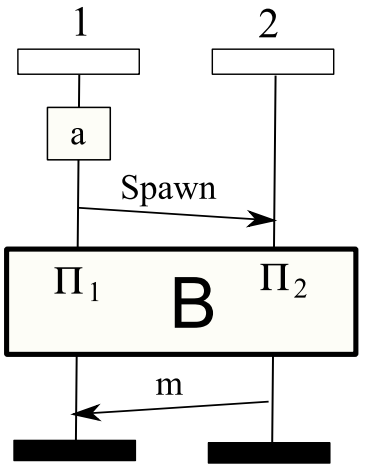
Axiom



MSC Grammars

Derivations:

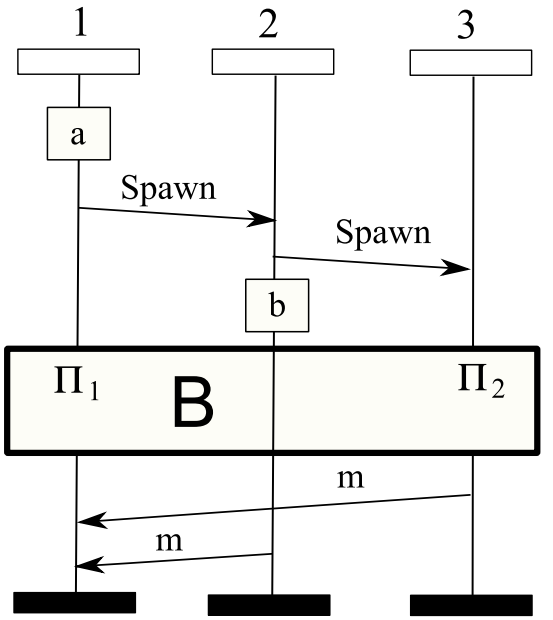
... as usual but writing MSCs instead of words/hypergraphs



MSC Grammars

Derivations:

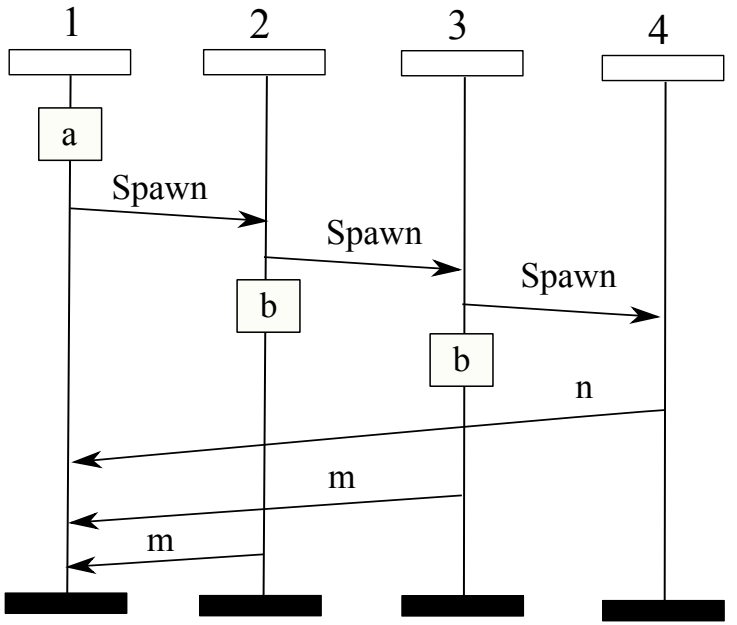
... as usual but writing MSCs instead of words/hypergraphs



MSC Grammars

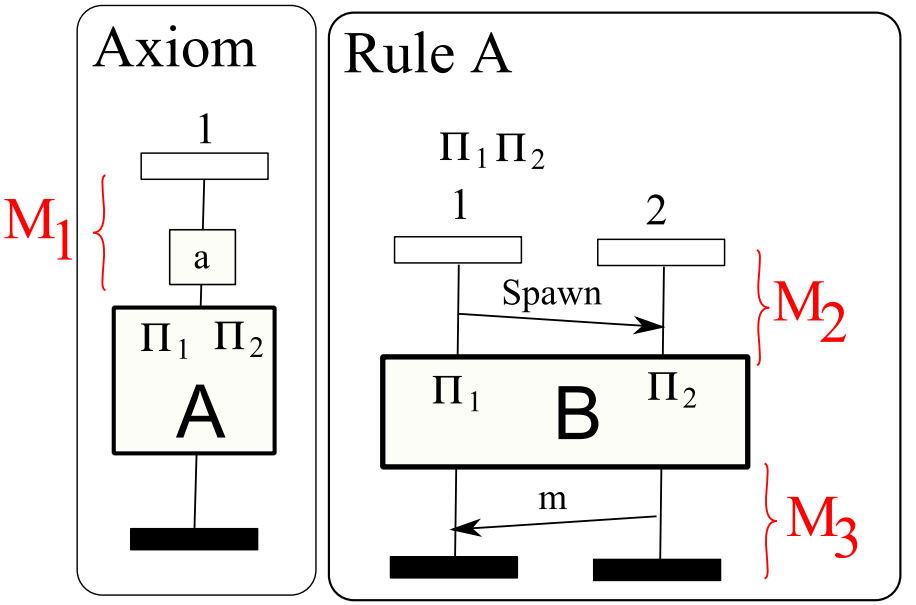
Derivations:

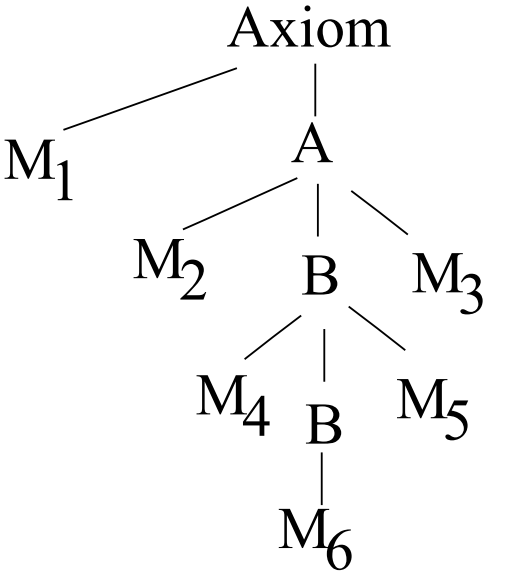
... as usual but writing MSCs instead of words/hypergraphs



MSC Grammars

Parse Tree





MSC Grammars

Language of an MSC grammar G :

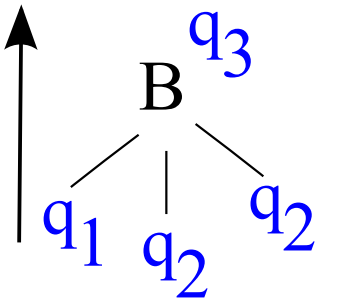
$$L(G) = \{M \in \mathbb{M} \mid \iota_G \Longrightarrow_G^* (M', \nu) \text{ for some } M' \cong M \text{ and } \nu\}.$$

All the MSCs that can be derived from the axiom of G.

Nb: $L(G)$ does not differentiate between isomorphic MSCs.

Tree automata

$TA = (Q, Q_F, \mathcal{F}, \delta)$



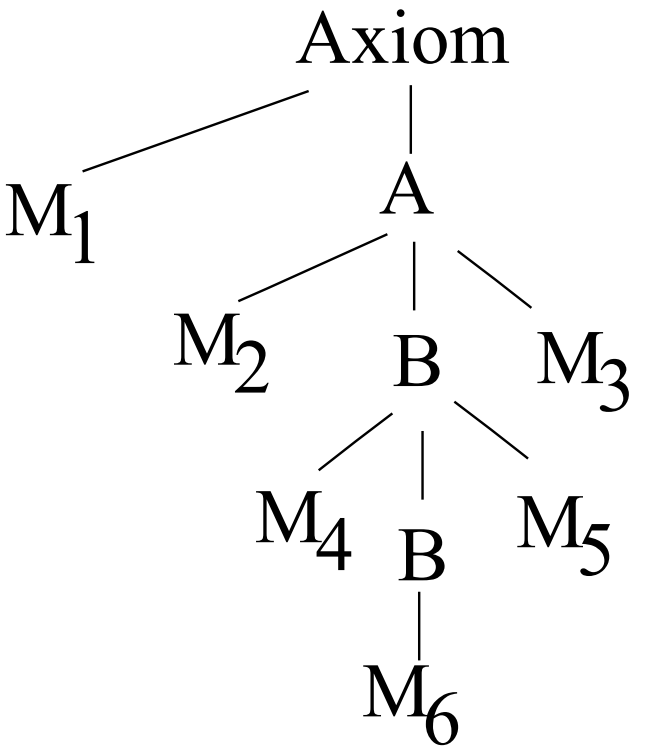
- Q : set of states
- $Q_F \subseteq Q$: set of final states
- $\mathcal{F}$ symbols (terminal and non terminal)
- $\delta \subseteq \mathcal{F} \times \bigcup_{1..K} Q^i \times Q$: transition relation

$$f(q_1(x_1), \ldots q_n(x_n)) \longrightarrow q(f(x_1), \ldots f(x_n))$$

A run of a TA on a tree $T = (N, \longrightarrow)$ is a mapping $r : N \longrightarrow Q$. r is successful if $r(root) \in Q_F$

TAs are recognizers for regular tree languages.

Tree automata

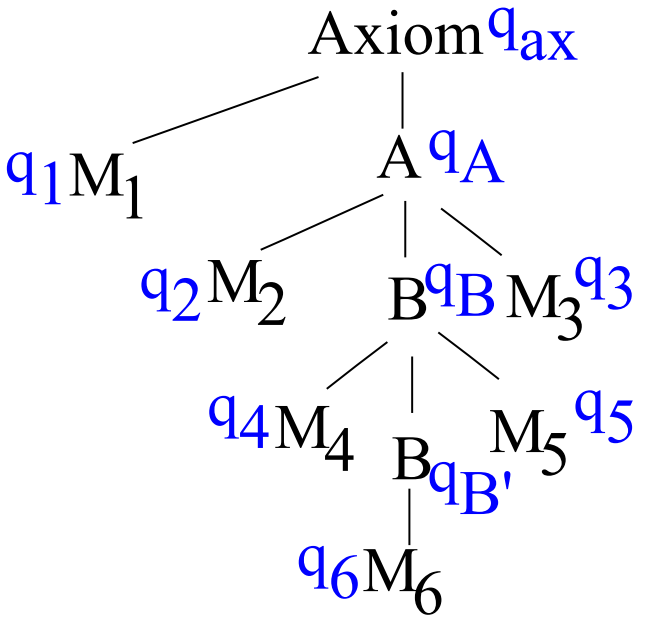


$$\mathcal{G} =$$

$$\begin{array}{lcl} Axiom & \longrightarrow & M_1.A \\ A & \longrightarrow & M_2.B.M_3 \\ B & \longrightarrow & M_4.B.M_5 \\ B & \longrightarrow & M_6 \end{array}$$

The derivation trees of a context free grammar are regular tree languages. The converse does not alway hold, except for *local tree languages*.

Tree automata



$TA_{\mathcal{G}}$=

$$
\begin{array}{lcl}
\bigcup M_i & \longrightarrow & q_i \\
Axiom(q_1(x), q_A(y)) & \longrightarrow & q_{Ax}(Axiom(x,y)) \\
A(q_2(x), q_B(y), q_3(z)) & \longrightarrow & q_A(A(x,y,z)) \\
B(q_4(x), q_B(y), q_5(z)) & \longrightarrow & q_B(B(x,y,z)) \\
B(q_6(x)) & \longrightarrow & q_{B'}(B(x))
\end{array}
$$

$$\mathcal{G} \longleftrightarrow TA_{\mathcal{G}}(local)$$

MSO over MSCs [Leucker& al02]

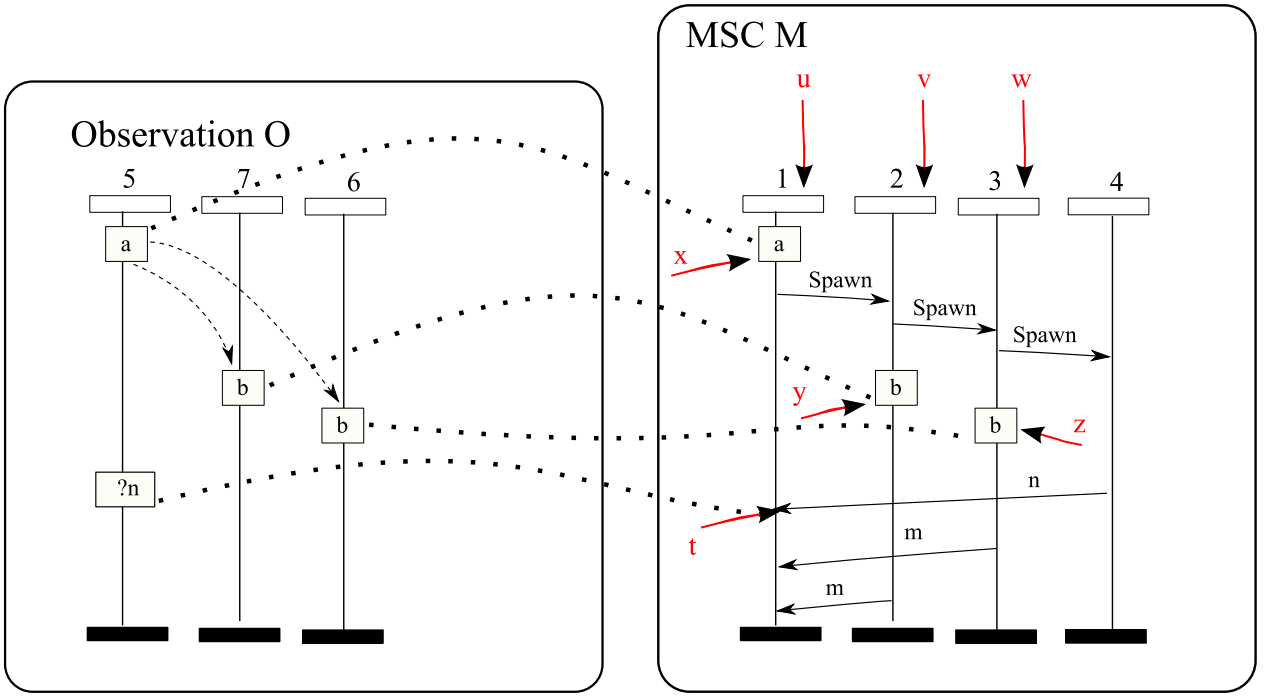
$$
\begin{array}{rll}
\varphi ::= & lab_a(x) & \text{x is an event labelled by a} \\
\mid & (u,x) \rightarrow (v,y) & \text{(x,y) is a message from u to v} \\
\mid & x \lessdot y & \text{x is the immediate predecessor of y on some process} \\
\mid & x \in X \mid u \in U & \\
\mid & \neg\varphi \mid \varphi_1 \wedge \varphi_2 & \\
\mid & \exists x\, \varphi \mid \exists X\, \varphi & \text{event/event set quantifier} \\
\mid & \exists u\, \varphi \mid \exists U\, \varphi & \text{process/process set quantifier}
\end{array}
$$

MSO over MSCs

Theorem 1 *Let O be an observation over a set of events $e_1 \ldots, e_n$, and M be a MSC and Σ_{Obs} be the observation alphabet. Then $O \triangleright_{\Sigma_{obs}} M$ if and only if $M \models \varphi_O$, where φ_O is the formula*

$$\begin{aligned}
\exists x_1, \ldots x_n, \quad & \bigwedge_{x \leq_O y} causalChain(x, y) \\
\wedge \quad & \bigwedge_{i \in 1..n} lab(x_i) = \lambda_O(e_i) \\
\wedge \quad & \bigwedge_{i \in 1..n} \nexists z, LocalPredecessor(x_i, z) \wedge lab(z) \in \Sigma_{Obs} \\
& \wedge z \notin x_1, \ldots x_n
\end{aligned}$$

MSO over MSCs



$$\varphi_O ::= \exists x, y, z, t, u, v, w,$$

$$lab_a(x) \wedge lab_b(y) \wedge lab_b(z) \wedge lab_{?n}(t)$$

$$\wedge LocalPredecessor(x, t)$$

$$\wedge (u, x) \leq (v, y) \wedge (u, x) \leq (w, z)$$

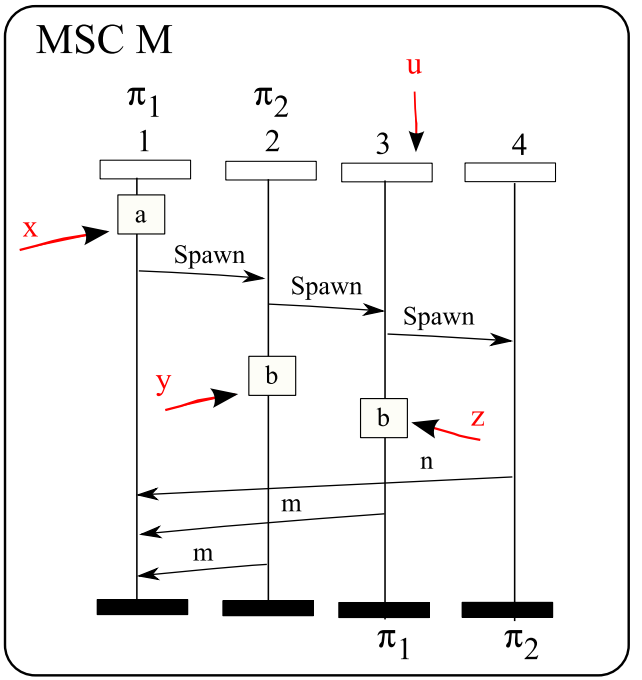
Results

Theorem 2 *MSO over MSCs is decidable for MSC grammars*

Corollary 1 *Diagnosis with MSC grammars is decidable*

Proof sketch [Leucker&al]

Interpreted tree automata that recognise parse trees



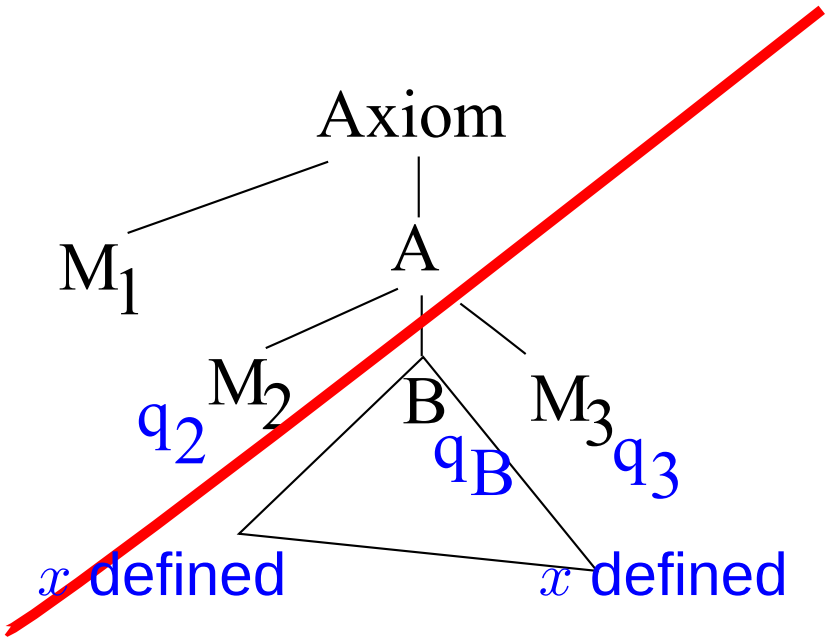
- guess $\gamma : V_{\varphi_O} \longrightarrow E_M \cup \mathcal{P}_M$
- infer $\pi_i \leq x$, $x \leq \pi_i'$, ...
- infer $\pi_i \leq \pi_i'$
- infer atoms of φ_O that hold at M

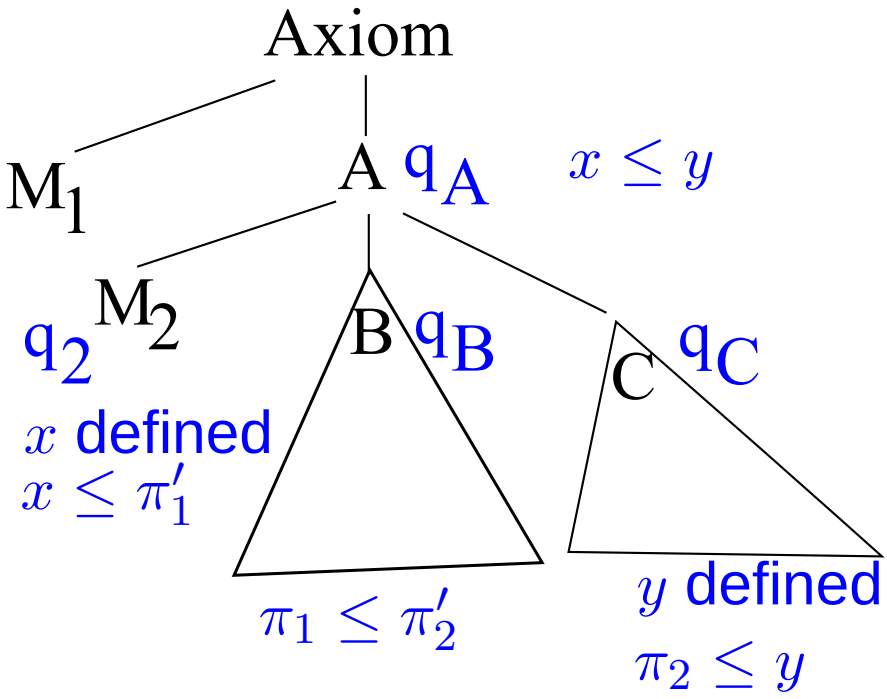
$$\exists x, y, z, u, lab_a(x), lab_b(y), lab_b(z)$$

$$\pi_1 \leq x, y, z \ \wedge \ \pi_2 \leq y, z \ \wedge \ x \leq \pi_1', \pi_2' \wedge z \leq \pi_1'$$

$$\pi_1 \leq \pi_1' \ \wedge \ \pi_1 \leq \pi_2' \ \wedge \ \pi_2 \leq \pi_1' \ \wedge \ \pi_2 \leq \pi_2'$$

Proof sketch [Leucker&al]





Accepting states : $q_{ax} \times \varphi$ such that $\varphi \Longrightarrow \varphi_O$

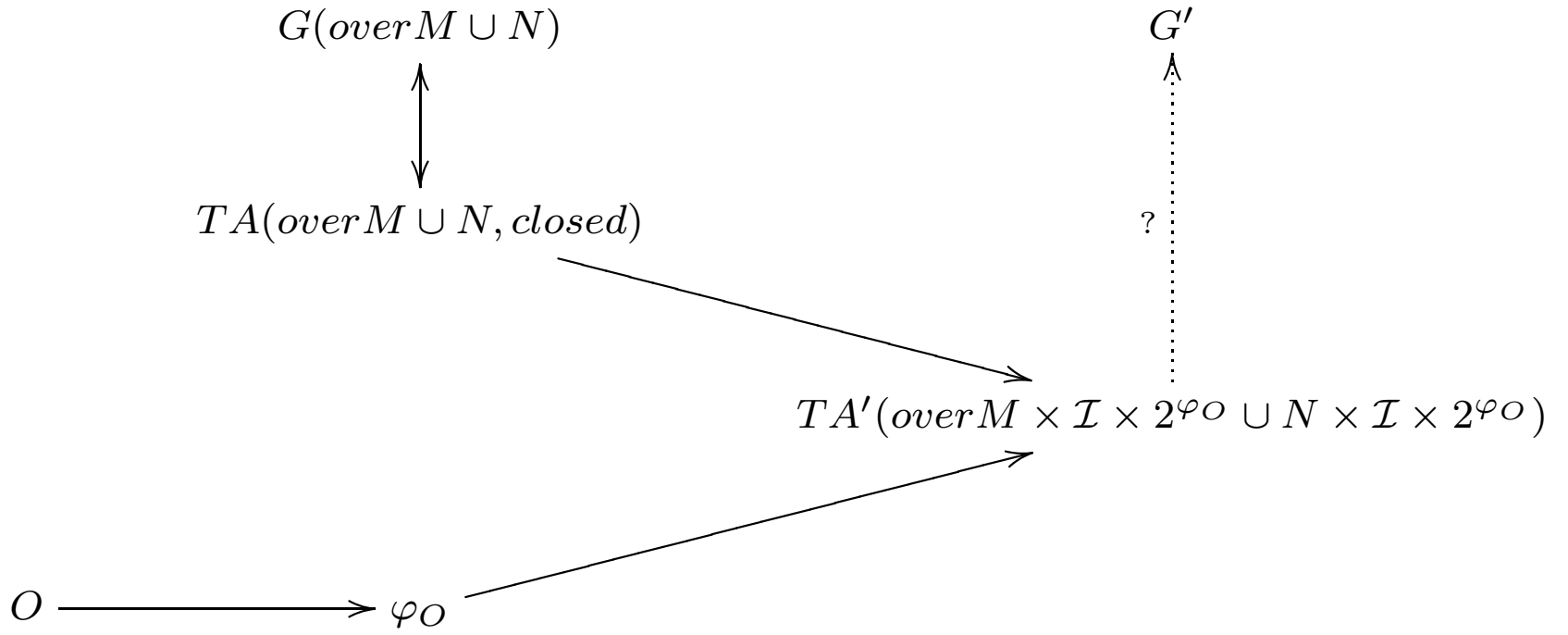
Diagnosis

Parse trees are decorated with:

- Interpretations over V_{φ_O} on leaves (MSCs) (finite)
- predicates denoting causal chains in a subtree (finite, involves identifiers or chosen proces in the subtree)
- Communication structure (mobility of processes identifiers)
- sub-formulae of φ_O that hold in the subtree

Tree automata transitions depend on consistence of labelings

Conclusion



We obtain a generator for all explanations of O

Conclusion

We have:

- Dynamic scenario model
- MSO/diagnosis decidable for it
- Generator comes for free as a consequence of decorated parse tree
- Compositionnality comes for free as a consequence of embeddings properties (NB : $\neq$ obs. , same grammar)

Future Work

Not a surprising result:

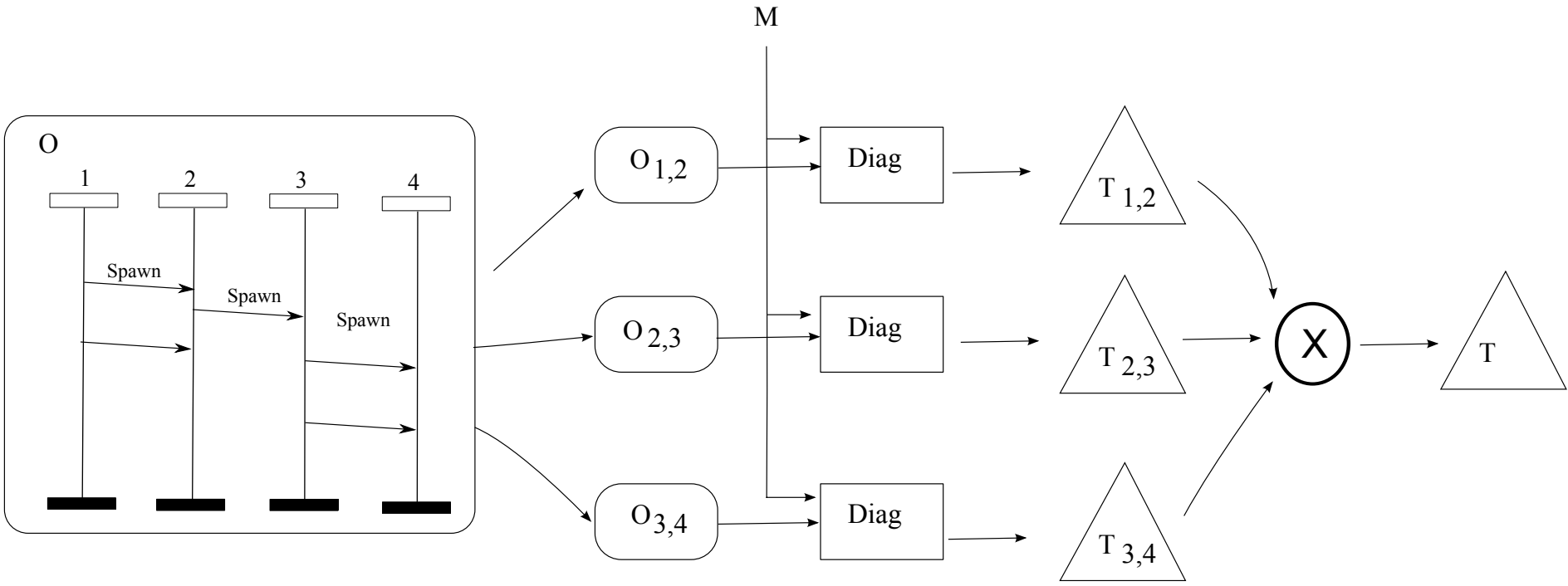
- MSC Grammars are graph grammars : all results apply [Courcelle]
- also subset of Dynamic MSCs [Leucker]
- Decidability only : MSO usually means exponential blowup !

Future Work

... that gives clues for efficient algorithms

- MSC Grammars as Hyperedge replacement + activation rule
- Not any MSO formula, not any kind of graph :
  - use the information on the model to avoid useless transitions in the TA
  - formula $\approx$ looking a sequence on each process, plus some inter process ordering ?

On compositionality



MSC grammars (Formal defs)

Definition 1 *A* (dynamic) MSC grammar *is a quadruple* $G = (\Pi, \mathcal{N}, S, \longrightarrow)$ *where*

- Π *and* $\mathcal{N}$ *and are nonempty finite sets of* process identifiers *and* non-terminals*, respectively,*
- $S \in \mathcal{N}$ *is the* start non-terminal*, and*
- $\longrightarrow$ *is a finite set of* rules

MSC Grammars (Formal defs)

Definition 2 *A rule is a triple* $r = (A, \alpha, f)$ *with*

- $A \in \mathcal{N}$ *non-terminal,*
- α *expression over* $\mathcal{N}$ *and* Π
- $f : Free(\alpha) \rightarrow \Pi$ *associates process identifiers to free processes of* α*.*

We may write $A \longrightarrow_f \alpha$*.*

MSO over MSCs

Causal chain : folklore [Madhusudan&al05]

$$\begin{aligned} causalChain(x,y) ::= \quad & \exists X, x \in X, y \in X, \\ & \forall Y \subset X, \exists x' \in max(X \setminus Y), \\ & closed(Y) \Longrightarrow \exists y' \in min(Y), \\ & x' \lessdot y' \vee x \longrightarrow y \end{aligned}$$

$$closed(Y) ::= \quad \nexists x, \nexists y \in Y, (x \lessdot y \vee x \longrightarrow y) \wedge x \notin Y$$

MSC grammars

MSC Grammars vs *Dynamic MSCs* [Leucker& al02]

- Dyn. MSCs more expressive than MSC grammars
- $L(G)$ has to be evaluated recursively (LR in MSC grammars)
- Implementation model for MSC grammars

Questions on MSC grammars

- $L(G) = \emptyset?$
- Realizability
- Implementation