Post hoc sed non propter hoc

or, Why You Should Think About Causality

Stefan Haar

INRIA and LSV
CNRS & ENS Cachan

Seminar SystemX, Sep 9, 2016

1. Why Causality ?
   - Example: Fault Diagnosis in Telecommunications
   - So, what is needed ?
2. Discrete Events, Partially ordered: Occurrence nets
   - Testing Causality
   - Diagnosis with Concurrency
3. Beyond Precedence
   - What Concurrency can reveal
   - Weak Diagnosis
4. Conclusion

Post hoc sed non propter hoc

1 Why Causality ?
- Example: Fault Diagnosis in Telecommunications
- So, what is needed ?

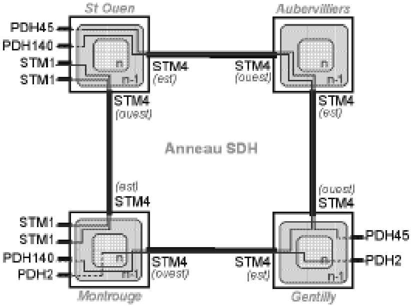
2 Discrete Events, Partially ordered: Occurrence nets

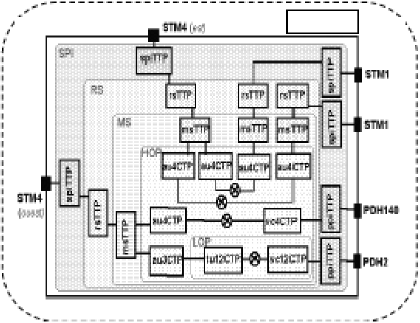
3 Beyond Precedence

4 Conclusion

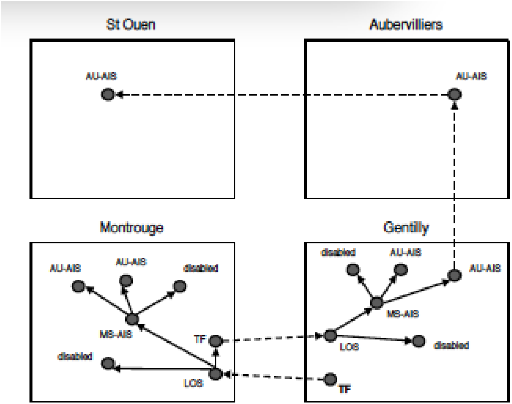
Fault Diagnosis in Telecommunications

Supervision SDH ring (Benveniste, Fabre, Haar et al, 2001 etc)



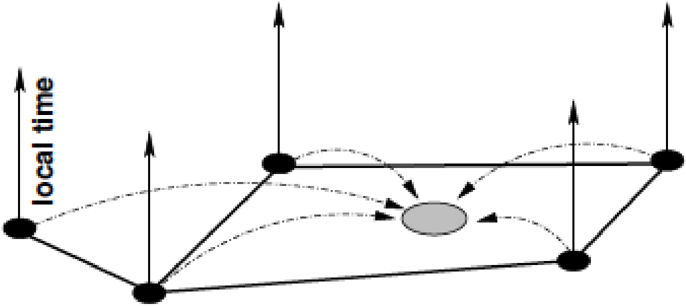


Telecom Supervision: Fault Propagation

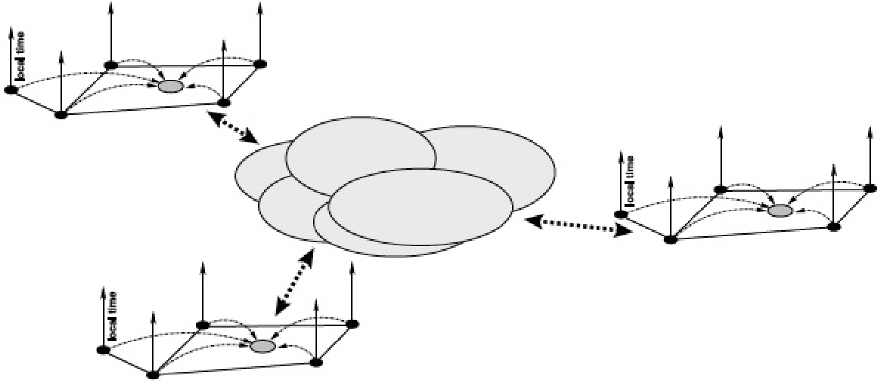


... but one observes only dots, not arrows

Asynchrony between occurrence and observation



Asynchrony + Distribution



The *Post hoc ergo propter hoc* fallacy

Description (from changingminds.org)

- If X follows Y, then X is caused by Y. (The sequence of things proves cause.)

Examples

- The man pulled out a gun. A shot was fired. Therefore the man fired the shot.
- You used the telephone and then it stopped working. You broke the phone.
- I am feeling very unwell. It must have been the meal last night.

Discussion

- Just because something follows something else, this is not sufficient evidence to prove true cause and effect. This temporal relationship may simply be coincidence.
- Coincidence is often related to superstition – hence saying 'bless you' when someone sneezes (it is assumed that sneezing lays a person open to spiritual attack) or throwing salt over your shoulder when you spill it (it is assumed to cause bad luck otherwise).

So, what is needed ?

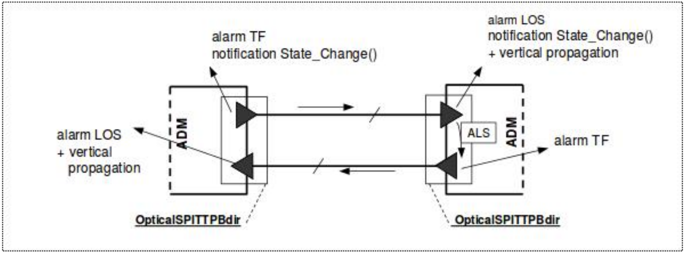
Observation is not enough

- Confront observations with a model of possible behaviours
- The model should contain all the information on causal dependencies, and nothing else

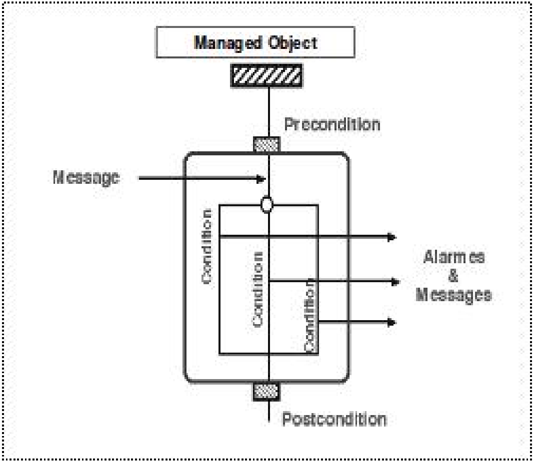
Modelling abstraction

- Separate the crucial functionalities and model only them
  - In the TIC example: ignore traffic, focus on fault propagation
- Drop non-crucial quantities (yes, including time stamps !)
- Be skeptical about time and aware of space
- Post hoc $\not\rightarrow$ propter hoc:
  - Partial order of causality vs total order of time

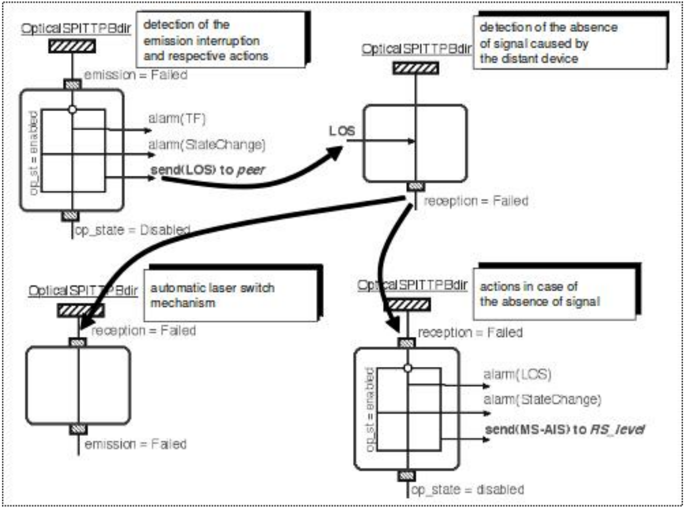
Example cont'd: SDH Laser Failure



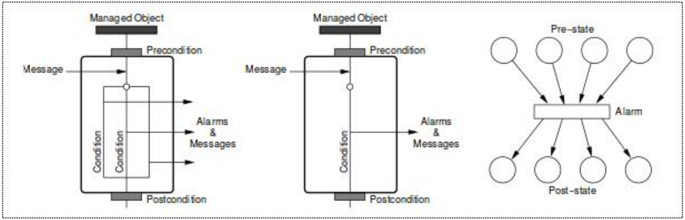
Generic model for Managed Object



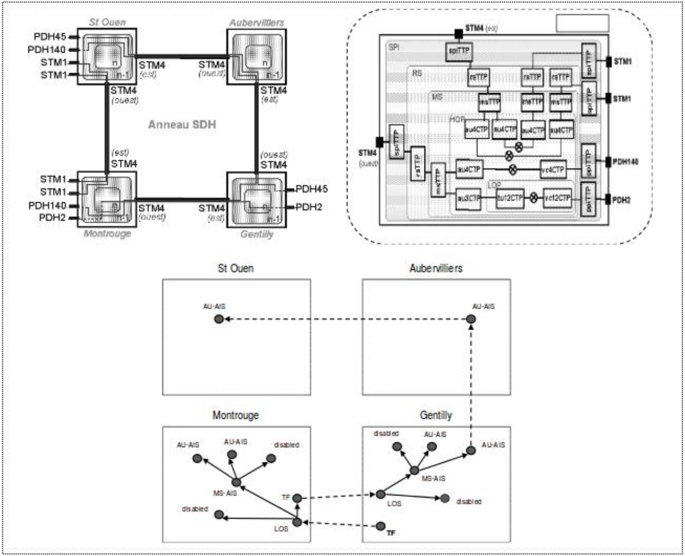
Combining Objects in Scenarios



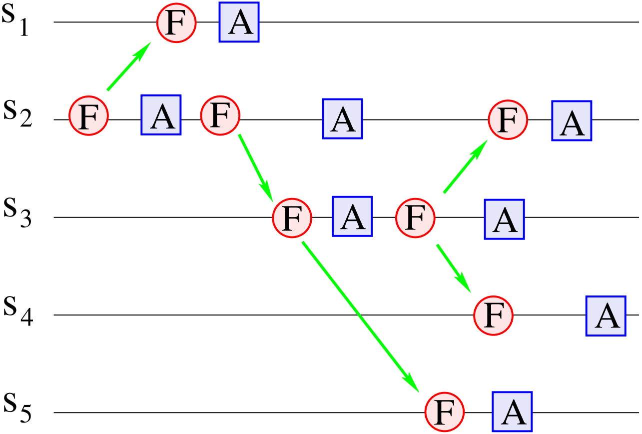
From Scenarios to Petri Nets



From such situations ...



… retain such pictures and analyse them !



Correlate observation with causal model

- will use *Petri nets* with *partial order semantics*

Post hoc sed non propter hoc

1 Why Causality ?

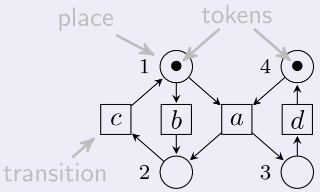
2 Discrete Events, Partially ordered: Occurrence nets
- Testing Causality
- Diagnosis with Concurrency

3 Beyond Precedence

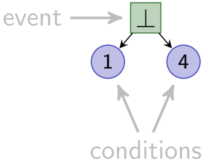
4 Conclusion

Petri nets, Processes, Branching Processes and Unfoldings

Petri net:

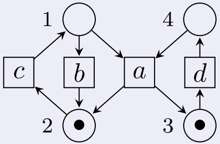


Process: representation of a non-sequential run as a partial order.

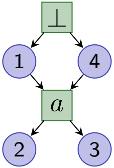


Petri nets, Processes, Branching Processes and Unfoldings

Petri net:

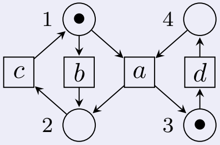


Process: representation of a non-sequential run as a partial order.

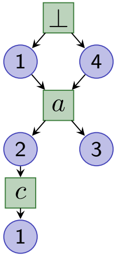


Petri nets, Processes, Branching Processes and Unfoldings

Petri net:

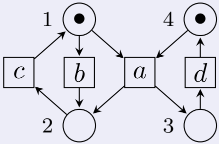


Process: representation of a non-sequential run as a partial order.

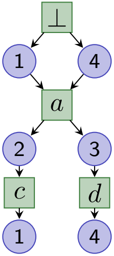


Petri nets, Processes, Branching Processes and Unfoldings

Petri net:

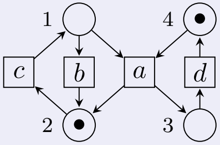


Process: representation of a non-sequential run as a partial order.

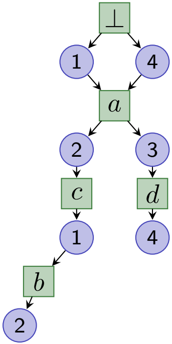


Petri nets, Processes, Branching Processes and Unfoldings

Petri net:

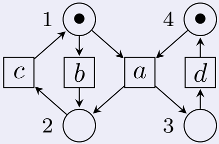


Process: representation of a non-sequential run as a partial order.



Petri nets, Processes, Branching Processes and Unfoldings

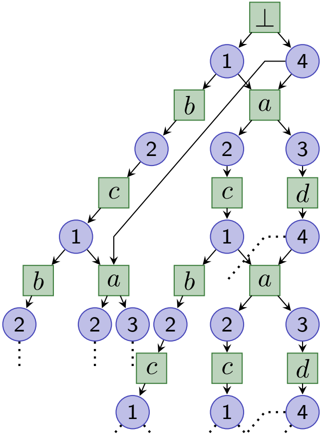
Petri net:



Process: representation of a non-sequential run as a partial order.

Branching process: representation of several runs.

Unfolding: maximal branching process.

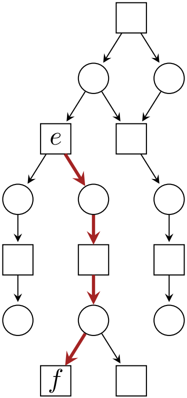


Nets and Structural Relations

The structure of a net induces three relations over its nodes:

Causality $\leq$

$e \leq f \quad \overset{def}{\Leftrightarrow} \quad e\ F^*\ f$ (directed path from e to f)



$e \leq f$

Nets and Structural Relations

The structure of a net induces three relations over its nodes:

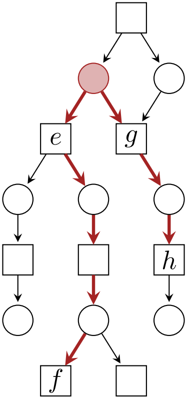
Causality $\leq$

$e \leq f \quad \overset{def}{\Leftrightarrow} \quad e\ F^*\ f$ (directed path from e to f)

Conflict $\#$

$e \#_d g \overset{def}{\Leftrightarrow} \quad e \neq g \wedge {}^\bullet e \cap {}^\bullet g \neq \emptyset$

$f \# h \overset{def}{\Leftrightarrow} \quad \exists e \leq f, g \leq h : e \#_d g$



$e \#_d g$

$f \# h$

Nets and Structural Relations

The structure of a net induces three relations over its nodes:

Causality $\leq$

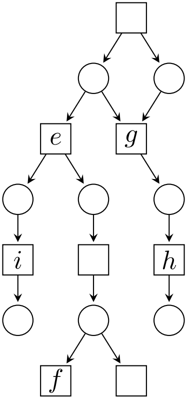
$e \leq f \quad \overset{def}{\Leftrightarrow} \quad e\ F^*\ f$ (directed path from e to f)

Conflict $\#$

$$e \#_d g \overset{def}{\Leftrightarrow} e \neq g \wedge {}^\bullet e \cap {}^\bullet g \neq \emptyset$$
$$f \# h \overset{def}{\Leftrightarrow} \exists e \leq f, g \leq h : e \#_d g$$

Concurrency co

$$f\ co\ i \overset{def}{\Leftrightarrow} \neg(i \# f) \wedge \neg(i \leq f) \wedge \neg(f \leq i)$$



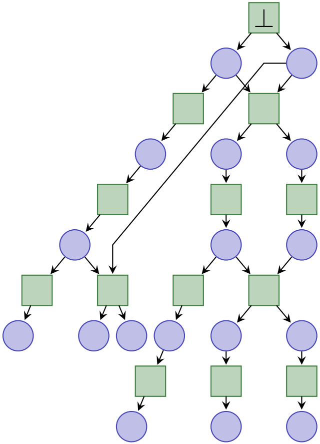
$f\ co\ i$

Occurrence Nets [Nielsen, Plotkin, Winskel, 1980]

Definition (Occurrence net)

An *occurrence net* (ON) is a net (B, E, F) where B and E are the sets of *conditions* and *events*, and which satisfies:

1. no self-conflict,
2. acyclicity
3. finite causal pasts: $\forall e \in E$, $\lceil e \rceil \stackrel{def}{=} \{e' : \ e' \leq e\}$ is finite.
4. no backward branching for conditions,
5. $\perp \in E$ is the only $\leq$-minimal node (event $\perp$ creates the initial conditions).



Configurations and Runs

Definitions (Configurations and Runs of an ON)

A *configuration* is a set ω of events which is

- causally closed: $\forall e \in \omega, \lceil e \rceil \subseteq \omega$,
- conflict free: $\forall e \in \omega, \#[e] \cap \omega = \emptyset$.

A run is a *maximal* (w.r.t. $\subseteq$) configuration.

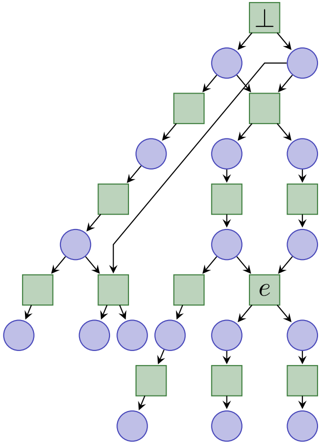
Notation

Ω denotes the set of *maximal runs*.

Interpretation

Ω gives exactly the *weakly fair* (nonsequential) executions:

- No transition remains enabled for ever (i.e. without firing, or being disabled by a conflicting transition): *weak fairness*



Configurations and Runs

Definitions (Configurations and Runs of an ON)

A *configuration* is a set ω of events which is

- causally closed: $\forall e \in \omega, \lceil e \rceil \subseteq \omega$,
- conflict free: $\forall e \in \omega, \#[e] \cap \omega = \emptyset$.

A run is a *maximal* (w.r.t. $\subseteq$) configuration.

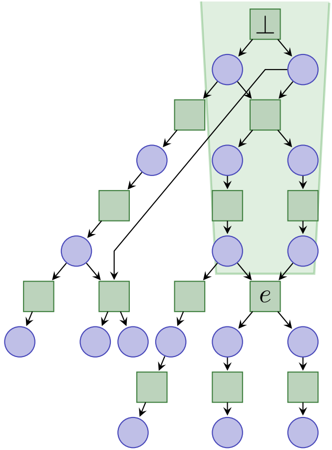
Notation

Ω denotes the set of *maximal runs*.

Interpretation

Ω gives exactly the *weakly fair* (nonsequential) executions:

- No transition remains enabled for ever (i.e. without firing, or being disabled by a conflicting transition): *weak fairness*



Configurations and Runs

Definitions (Configurations and Runs of an ON)

A *configuration* is a set ω of events which is

- causally closed: $\forall e \in \omega, \lceil e \rceil \subseteq \omega$,
- conflict free: $\forall e \in \omega, \#[e] \cap \omega = \emptyset$.

A run is a *maximal* (w.r.t. $\subseteq$) configuration.

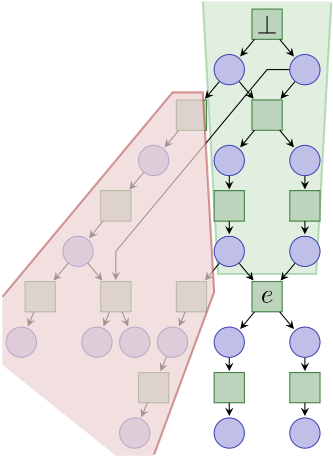
Notation

Ω denotes the set of *maximal runs*.

Interpretation

Ω gives exactly the *weakly fair* (nonsequential) executions:

- No transition remains enabled for ever (i.e. without firing, or being disabled by a conflicting transition): *weak fairness*



Using Occurrence Nets

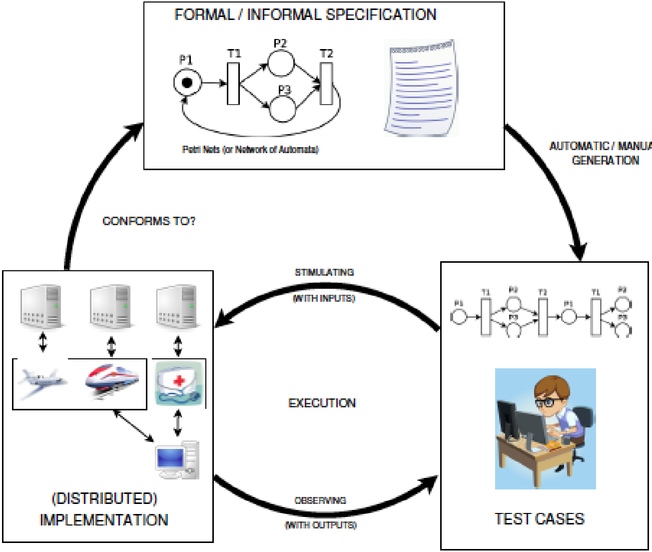
A Data Structure representing Causality

- No arrow chain, no causal link (see below however)
- Concurrency as the dual of causality

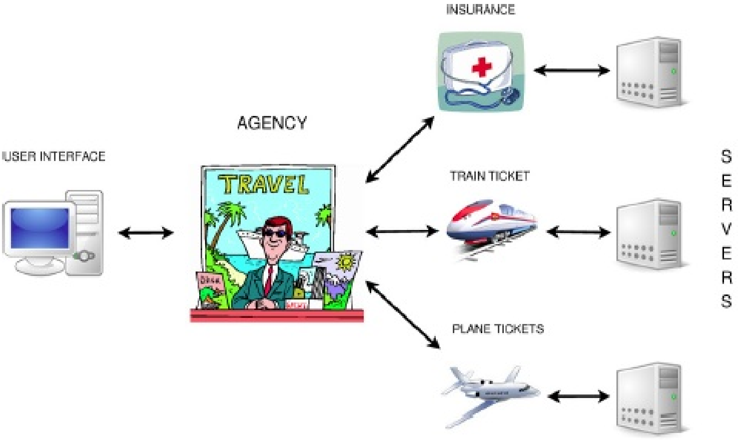
Fields

- Testing (next)
- Diagnosis for Telecom supervision (later)
- Verification of programs (not here)
- Systems Biology (not here)
- ... (you name it)

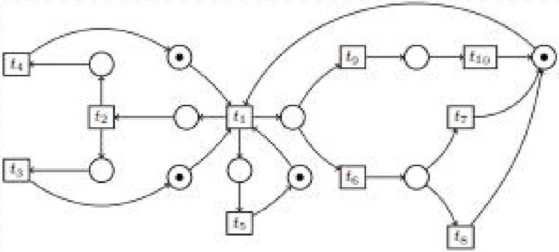
Testing causality (H. Ponce de Leon et al, since $\sim$ 2011)



Example


INSURANCE
AGENCY
USER INTERFACE
TRAVEL
TRAIN TICKET
PLANE TICKETS
S E R V E R S

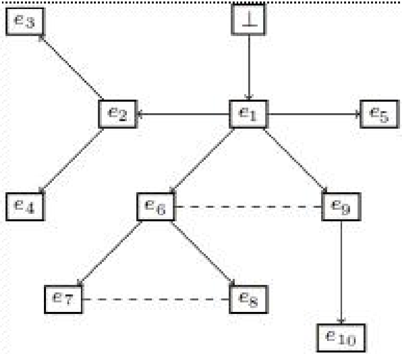
PN model ...



t	$\lambda(t)$
t_1	$?login$
t_2	$?ms$
t_3	$!price_i$
t_4	$!data_i$
t_5	$!us\_data$

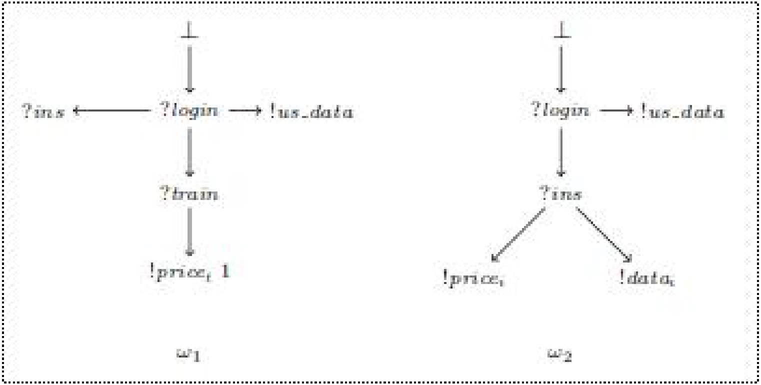
t	$\lambda(t)$
t_6	$?train$
t_7	$!price_t\ 1$
t_8	$!price_t\ 2$
t_9	$?plane$
t_{10}	$!price_p$

… and its partial order of events



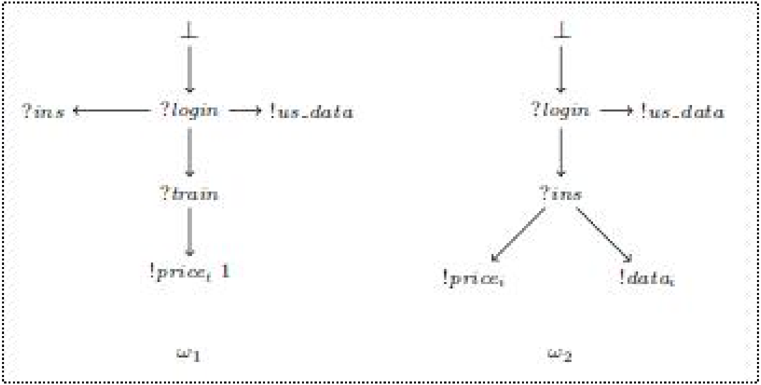
e	$\varphi(e)$	$\lambda(e)$
e_1	t_1	$?login$
e_2	t_2	$?ins$
e_3	t_3	$!price_i$
e_4	t_4	$!data_i$
e_5	t_5	$!us\_data$
e_6	t_6	$?train$
e_7	t_7	$!price_t\ 1$
e_8	t_8	$!price_t\ 2$
e_9	t_9	$?plane$
e_{10}	t_{10}	$!price_p$

Traces



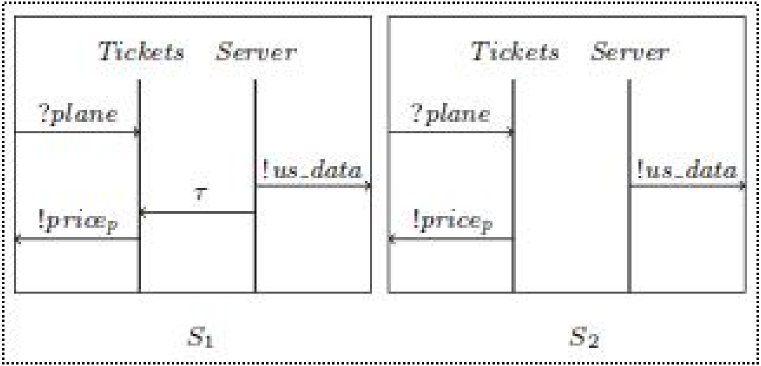
Use for checking conformance with specified causality (and concurrency)

Test Cases



Use for checking conformance with specified causality (and concurrency)

Concurrency in Specification

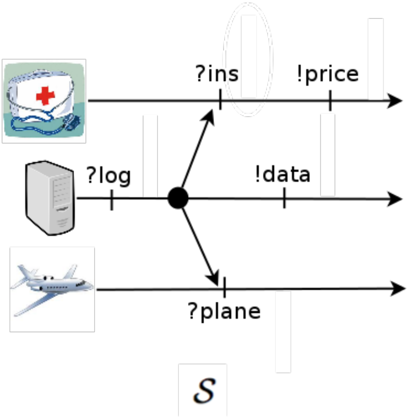


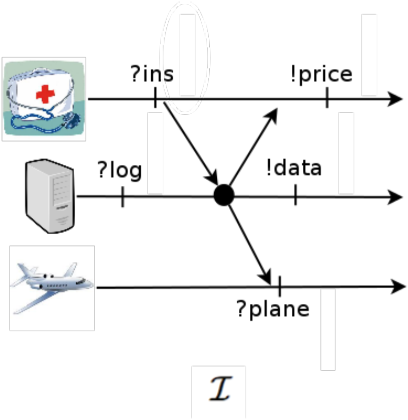
Causality in Testing

Asynchronous Testing with PNs

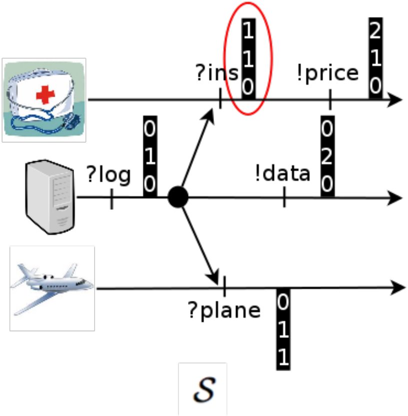
- Avoids concurrency pitfalls (observability etc) known from multi-channel testing over FSM
- Can:
  - test for respect of causal dependencies from specifications
  - tolerate ordering of some concurrently specified events (*'don't care'*-concurrency)
  - test for respect of *intended* or *strong* concurrency

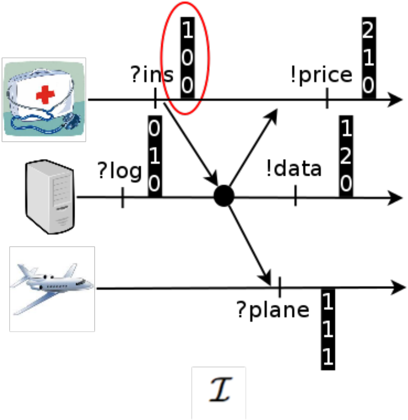
Testing for concurrency: Vector clocks





Testing for concurrency: Vector clocks



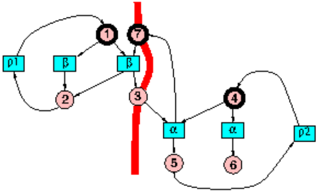


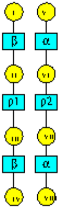
Diagnosis with Concurrency

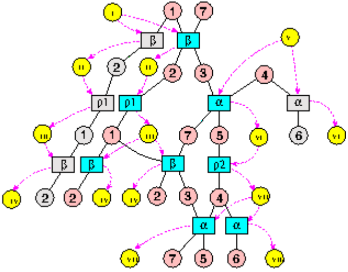
Back to the Telecom Example

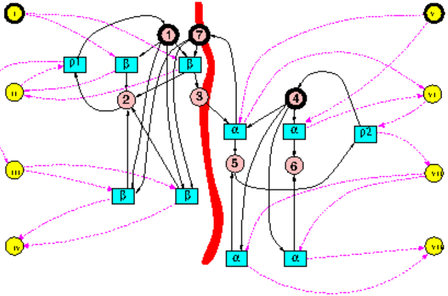
- PN model of fault propagation
- one or more strings of alarms
- correlate via product
- filter out *partially ordered* explanations (configurations)
- Supposing Diagnosability (another long subject...)

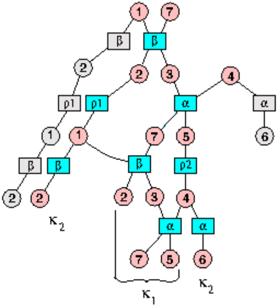
Telecom Supervision











Diagnosis with unfoldings

A Success Story

- Causal/Concurrency model adequat
- ... and efficient: store only partial order, not all its interleavings
- Scales up wrt growing number of parts
- Allows distribution
- Run successfully on ring supervision platform
- Handles centralized and distributed monitoring

Can do more

- Partial order structures reveal dependencies and implication across parallel processes
- Exploited in *weak diagnosis* $\rightarrow$ NEXT

Post hoc sed non propter hoc

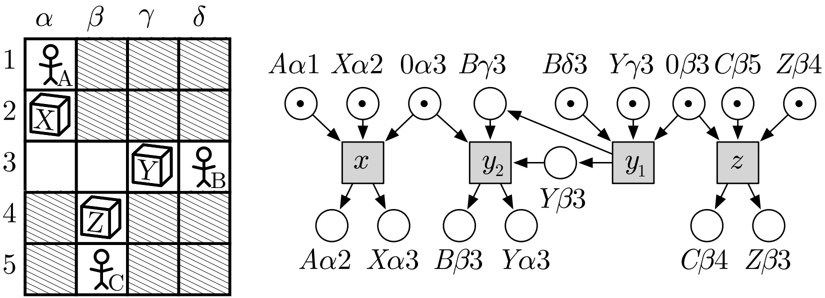
1 Why Causality ?

2 Discrete Events, Partially ordered: Occurrence nets

3 Beyond Precedence
- What Concurrency can reveal
- Weak Diagnosis

4 Conclusion

Some actions reveal one another



z prevents y_1 ... and therefore makes x inevitable:

$$z \textbf{ reveals } x \quad : \quad z \triangleright x$$

Reveals Relation [Haar, 2010]

Definition (Reveals relation $\rhd$)

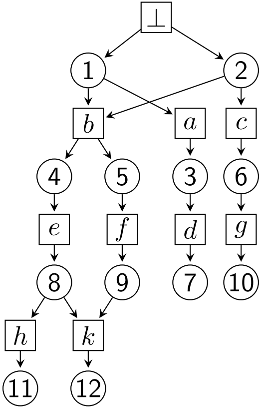
Event e reveals event f, written $e \rhd f$, iff $\forall \omega \in \Omega, (e \in \omega \Rightarrow f \in \omega)$.

Causal closure

$\forall x, y \in E, x \leq y \Rightarrow y \rhd x$

$d \rhd a$,
$h \rhd \perp$,
$a \rhd d$
because of weak fairness,
$a \rhd c$
because for any maximal run ω,

$$\begin{aligned} a \in \omega \quad &\Rightarrow \quad b \notin \omega \\ &\Rightarrow \quad c \in \omega \text{ (weak fairness)} \end{aligned}$$



Reveals Relation [Haar, 2010]

Definition (Reveals relation $\triangleright$)

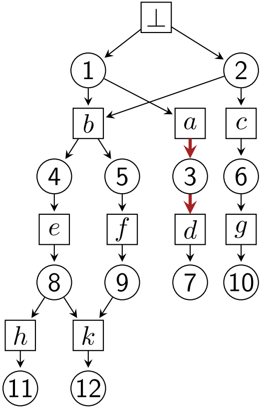
Event e *reveals event* f, *written* $e \triangleright f$, *iff* $\forall \omega \in \Omega, (e \in \omega \Rightarrow f \in \omega)$.

Causal closure

$\forall x, y \in E, x \leq y \Rightarrow y \triangleright x$

$d \triangleright a$,
$h \triangleright \perp$,
$a \triangleright d$
because of weak fairness,
$a \triangleright c$
because for any maximal run ω,

$$\begin{aligned} a \in \omega &\Rightarrow b \notin \omega \\ &\Rightarrow c \in \omega \text{ (weak fairness)} \end{aligned}$$



Reveals Relation [Haar, 2010]

Definition (Reveals relation $\triangleright$)

Event e reveals event f, written $e \triangleright f$, iff $\forall \omega \in \Omega, (e \in \omega \Rightarrow f \in \omega)$.

Causal closure

$\forall x, y \in E, x \leq y \Rightarrow y \triangleright x$

$d \triangleright a$,

$h \triangleright \perp$,

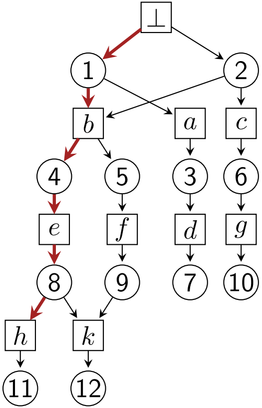
$a \triangleright d$

because of weak fairness,

$a \triangleright c$

because for any maximal run ω,

$$\begin{aligned} a \in \omega \quad &\Rightarrow \quad b \notin \omega \\ &\Rightarrow \quad c \in \omega \text{ (weak fairness)} \end{aligned}$$



Reveals Relation [Haar, 2010]

Definition (Reveals relation $\rhd$)

Event e reveals event f, written $e \rhd f$, iff $\forall \omega \in \Omega, (e \in \omega \Rightarrow f \in \omega)$.

Causal closure

$\forall x, y \in E, x \leq y \Rightarrow y \rhd x$

$d \rhd a$,

$h \rhd \bot$,

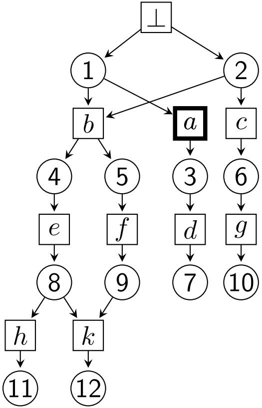
$a \rhd d$

because of weak fairness,

$a \rhd c$

because for any maximal run ω,

$$\begin{aligned} a \in \omega \quad &\Rightarrow \quad b \notin \omega \\ &\Rightarrow \quad c \in \omega \text{ (weak fairness)} \end{aligned}$$



Reveals Relation [Haar, 2010]

Definition (Reveals relation $\triangleright$)

Event e reveals event f, written $e \triangleright f$, iff $\forall \omega \in \Omega, (e \in \omega \Rightarrow f \in \omega)$.

Causal closure

$\forall x, y \in E, x \leq y \Rightarrow y \triangleright x$

$d \triangleright a$,

$h \triangleright \bot$,

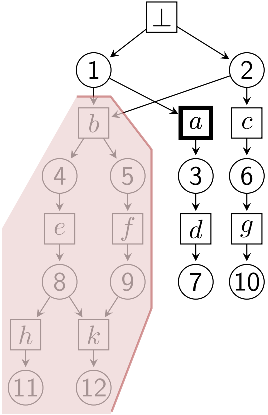
$a \triangleright d$

because of weak fairness,

$a \triangleright c$

because for any maximal run ω,

$$a \in \omega \quad \Rightarrow \quad b \notin \omega$$
$$\Rightarrow \quad c \in \omega \text{ (weak fairness)}$$



Reveals Relation [Haar, 2010]

Definition (Reveals relation $\triangleright$)

Event e reveals event f, written $e \triangleright f$, iff $\forall \omega \in \Omega, (e \in \omega \Rightarrow f \in \omega)$.

Lemma

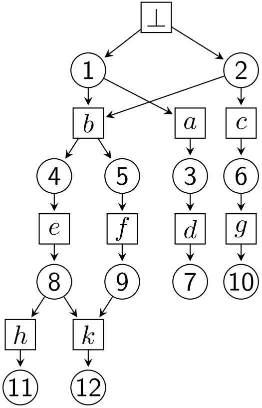
Lemma: Characterization of Ω by $\#$ A set of events ω is a maximal run iff

$$\forall a \in E, a \notin \omega \Leftrightarrow \#[a] \cap \omega \neq \emptyset$$

where $\#[e] \stackrel{def}{=} \{f \in E \mid f \# e\}$.

Characterization of $\triangleright$ by $\#$

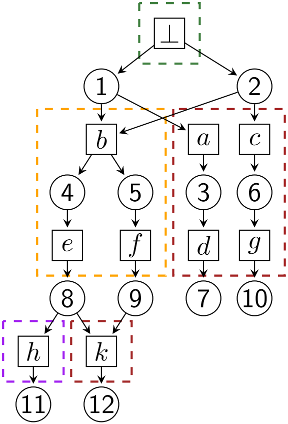
$\forall e, f \in E$, $e \triangleright f \Leftrightarrow \#[f] \subseteq \#[e]$
i.e. any event that could prevent the occurrence of f is prevented by the occurrence of e.



Facets Abstraction [H2010,BCH2011]

Definition (Facets)

A facet of an ON is an equivalence class of $\sim = \triangleright \cap \triangleright^{-1}$.



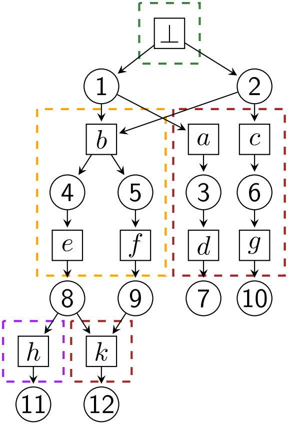
Facets Abstraction [H2010,BCH2011]

Definition (Facets)

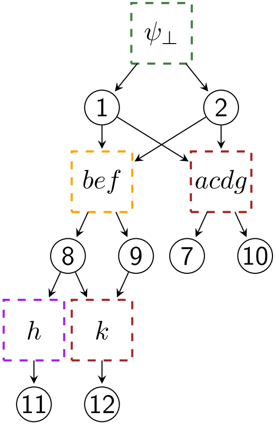
A facet of an ON is an equivalence class of $\sim = \triangleright \cap \triangleright^{-1}$.

[

Reduced ON] Contracting Facets yield (bigger) events for a reduced ON
reduced ON is an ON (B, Ψ, F) such that $\forall \psi_1, \psi_2 \in \Psi$, $\psi_1 \sim \psi_2 \Leftrightarrow \psi_1 = \psi_2$.

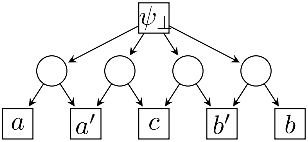


facets can be contracted into events



Concurrency vs Logical Independency [BCH2011]

- $\#$, $\leq$ and co are mutually exclusive.



Structural relations and logical dependencies

- $a \# b \Leftrightarrow$ for any run ω, $\{a, b\} \not\subseteq \omega$.
- $a \leq b \Rightarrow$ for any run ω, $b \in \omega \Rightarrow a \in \omega$ $(b \triangleright a)$,
- Does a co b mean a and b are logically independent ?

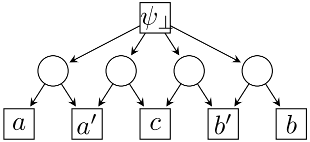
  No, they can be related by $\triangleright$.

c co a and $c \triangleright a$

a co b and a ind b.

Concurrency vs Logical Independency [BCH2011]

- $\#$, $\leq$ and co are mutually exclusive.



Structural relations and logical dependencies

- $a \# b \Leftrightarrow$ for any run ω, $\{a, b\} \not\subseteq \omega$.
- $a \leq b \Rightarrow$ for any run ω, $b \in \omega \Rightarrow a \in \omega$ $(b \triangleright a)$,
- Does a co b mean a and b are logically independent ?
  No, they can be related by $\triangleright$.

c co a and $c \triangleright a$
a co b and a ind b.

Independency relation ind

$$\forall a, b \in \Psi, \quad a \; ind \; b \quad \overset{def}{\Leftrightarrow} \quad \neg(a \# b) \wedge \neg(b \triangleright a) \wedge \neg(a \triangleright b)$$
$$\Leftrightarrow \quad a \; co \; b \wedge \neg(b \triangleright a) \wedge \neg(a \triangleright b)$$

- $\#$, $\triangleright$ and ind are also mutually exclusive.

Extended Reveals Relation

Definition (Extended reveals relation)

Let A, B be two sets of facets.
A reveals B, written $A \twoheadrightarrow B$, iff $\forall \omega \in \Omega, A \subseteq \omega \Rightarrow B \cap \omega \neq \emptyset$.

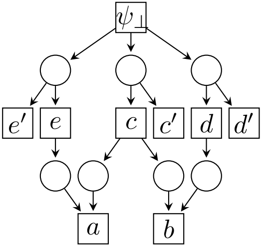
Properties

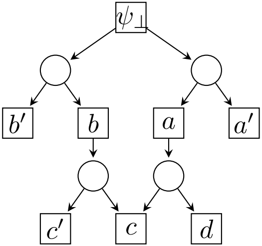
- $\{a\} \twoheadrightarrow \{b\} \Leftrightarrow a \triangleright b$
- Conflicts can be expressed with this extended reveals relation: $\{a, b\} \twoheadrightarrow \emptyset \Leftrightarrow a \# b$.

Extended Reveals Relation

Examples

$$A \twoheadrightarrow B \quad \Leftrightarrow \quad \forall \omega \in \Omega, A \subseteq \omega \Rightarrow B \cap \omega \neq \emptyset$$





$$\{c, e\} \twoheadrightarrow \{a\}$$
$$\{c, d, e\} \twoheadrightarrow \{a\}$$
$$\{e', e\} \twoheadrightarrow \emptyset$$

$$\{a, b\} \twoheadrightarrow \{c', c, d\}$$
$$\{a\} \twoheadrightarrow \{c, d\}$$
$$\emptyset \twoheadrightarrow \{a, a'\}$$

Diagnosis: Sequential Semantics Misses a Point

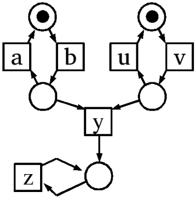
Suppose that

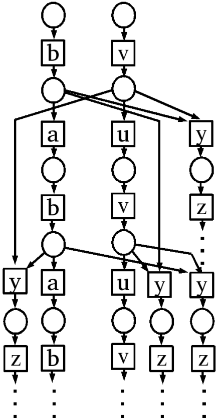
- $T_O = \{b, y\}$
- $\Phi = \{v\}$

v will be correctly diagnosed if y occurs. What if not ? If

$$bbbbbb\ldots$$

is observed, what do we infer about v ?



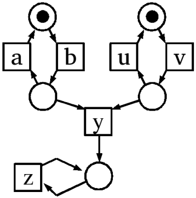


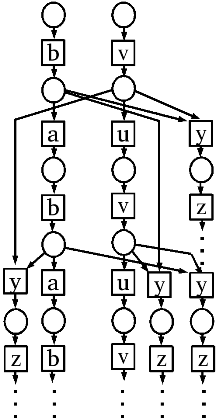
It's about weak fairness !

Still with

- $T_O = \{b, y\}$
- $\Phi = \{v\}$

the only way for the system to do b^ω is to be *unfair* to v: always enabled, never fired
HERE: diagnosis under weak fairness





Extended Reveals+Diagnosis

Application

- $A \twoheadrightarrow B$ iff ρ's *containing* A must *hit* B
- Used for *weak diagnosis*:
  Given an observation pattern α, are *all* weakly fair extensions of explanations of α faulty ?

Weak Diagnosis

Observation pattern α *weakly diagnoses* fault ϕ iff

$$C \in expl(\alpha) \quad \Rightarrow \quad C \rightharpoonup E_\phi$$

Weak Diagnosis

Spoilers

Let $t \in T$. The set of t's *spoilers* is

$$spoil(t) \stackrel{def}{=} \{t' \in T \mid {}^\bullet t' \cap {}^\bullet t \neq \emptyset\}.$$

Note : $t \in spoil(t)$!

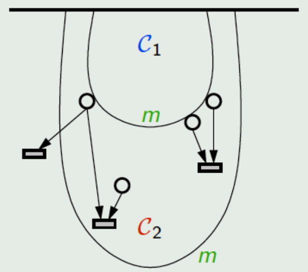
Weak Fairness

A run $\rho = M_0 t_1 M_1 t_2 \ldots$ is *weakly* fair iff every transition t enabled in some M_k is *disabled* in some M_{k+j}; that is, eventually one of $t's$ spoilers fires !

Lemma

There is ω weakly-fair and fault-free iff there are configurations $\mathcal{C}_1, \mathcal{C}_2$ such that:

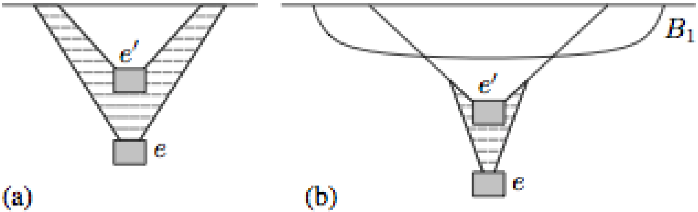
1. $\mathcal{C}_1 \subseteq \mathcal{C}_2$
2. $mark(\mathcal{C}_1) = mark(\mathcal{C}_2)$
3. $\mathcal{C}_1$ enables $e \Rightarrow spoilers(e) \cap \mathcal{C}_2 \neq \emptyset$
4. $\mathcal{C}_2$ is fault-free



Solving the weak diagnosis problem

Weak Diagnosis Problem

$$C \in expl(\alpha) \quad \overset{???}{\Longrightarrow} \quad C \twoheadrightarrow E_\phi \qquad (*)$$



- Take a *marking-complete* prefix B_1
- Stop unfolding at *sp-cutoff events* e, i.e. there is $e' < e$ s.th. , for $D \overset{def}{=} [e]\backslash[e']$,
  - $f({}^\bullet D \backslash D^\bullet) = f(D^\bullet \backslash {}^\bullet D)$ and $B_1 \cap {}^\bullet D = \emptyset$,

  I.e. e and e' spoil exactly the same events enabled by configurations from B_1.

Decision method

Prefixes needed

- P_α: contains all *succinct* explanations of α
- P^1: marking-complete
- P^2: contains all *non-sp-cutoffs*; $P^1 \sqsubseteq P^2$

ALL ARE FINITE !!

Encoding in SAT

$$config(l, \mathcal{P}) \stackrel{def}{=} \Big(\bigwedge_{e \in E} \bigwedge_{e' \in {}^{\bullet\bullet}e} (v^l_e \Rightarrow v^l_{e'}) \Big) \quad \wedge$$

$$\Big(\bigwedge_{c \in B, \{e_1, \ldots, e_n\} = c^\bullet} amo(v^l_{e_1}, \ldots, v^l_{e_n}) \Big) \quad \wedge \quad \Big(\bigwedge_{c \in B} v^l_c \Leftrightarrow \big(\bigwedge_{e \in {}^\bullet c} v^l_e \wedge \bigwedge_{e \in c^\bullet} \neg v^l_e \big) \Big)$$

- Similarly : configuration containment, reachability, enabling, spoiling, explanation,...
- Diagnosis checkable with SAT solvers

Post hoc sed non propter hoc

1 Why Causality ?

2 Discrete Events, Partially ordered: Occurrence nets

3 Beyond Precedence

4 Conclusion

Causality is more informative than time

Modelling abstraction

- In the causal partial order model, dependence relations are local
- Spurious 'ordering' from observation avoided
- Often: computation time + space gained
- Always: conceptual error avoided
- Causal precedence implies temporal one, but not the other way around
- Be skeptical about time and aware of space

Fields that should be causality-aware

- Telecom, Web services
- Supervision of Networks
- Forensics
- Business (and other) processes

A Partial To Do list (for research)

Diagnosis

- Pursue *active* diagnosis: if observation is insufficient, force more significant output
- Be sure to do *save* active diagnosis: don't force occurrence of a fault only so you can diagnose it
- Generalize reveals to probabilities

Process Mining

- Infer or improve causal models via log analysis
- Separate causal from spurious

Food for thought

Time $\leftrightarrow$ Causality ?

- Philosophical definitions of causality tend to use temporal precedence ;
- Conversely, time is captured via causal chains: ???
- Indirect causalities ('*reveals*') may transcend temporal orderings and jump between causal chains;
-
  - however, weak fairness was needed to capture them, i.e. a temporal property is at the heart of *reveals*;
  - moreover, thus far we need vector clocks to test strong concurrency;
- Maybe time and causality are inextricable ?
- Remark: do not confound causality and inference

Thanks

Thanks to

- Albert Benveniste, Eric Fabre and Armen Aghasaryan
- Agnes Madalinkski and Victor Khomenko
- Sandie Balaguer and Thomas Chatain
- Delphine Longuet and Hernán Ponce de León
- Stefan Schwoon, César Rodríguez and Christian Kern
- ...
- and last but not least you !

References

- H. Ponce de León, S. Haar and D. Longuet. Model-based Testing for Concurrent Systems: Unfolding-based Test Selection. Int. J.n Software Tools for Technology Transfer, 2015.
- V. Germanos, S. Haar, V. Khomenko and S. Schwoon. Diagnosability under Weak Fairness. ACM Transactions in Embedded Computing Systems 14(4:69), 2015.
- H. Ponce de León, S. Haar and D. Longuet. Model-Based Testing for Concurrent Systems with Labeled Event Structures. STVR 24(7), 558-590, 2014.
- S. Haar, C. Kern and S. Schwoon. Computing the Reveals Relation in Occurrence Nets. Theoretical Computer Science 493, pages 66-79, 2013.
- S. Balaguer, Th. Chatain and S. Haar. Building Occurrence Nets from Reveals Relations. Fundamenta Informaticae 123(3), pages 245-272, 2013.
- S. Haar. What topology tells us about diagnosability in partial order semantics. Discrete Event Dynamic Systems 22(4), pages 383-402, 2012.
- E. Fabre and A. Benveniste. Partial Order Techniques for Distributed Discrete Event Systems: why you can't avoid using them. *Discrete Event Dynamic systems* 2007.
- P. Baldan, Th. Chatain, S. Haar and B. König. Unfolding-based Diagnosis of Systems with an Evolving Topology. Information and Computation 208(10), pages 1169-1192, 2010.
- S. Haar. Types of Asynchronous Diagnosability and the Reveals-Relation in Occurrence Nets. IEEE Transactions on Automatic Control 55(10), pages 2310-2320, 2010.
- E. Fabre, A. Benveniste, S. Haar and C. Jard. Distributed monitoring of concurrent and asynchronous systems. Discrete Event Dynamic Systems 15(1), pages 33-84, 2005.
- A. Benveniste, E. Fabre, S. Haar and C. Jard. Diagnosis of Asynchronous Discrete Event Systems: A Net Unfolding Approach. IEEE Transactions on Automatic Control 48(5), pages 714-727, 2003.