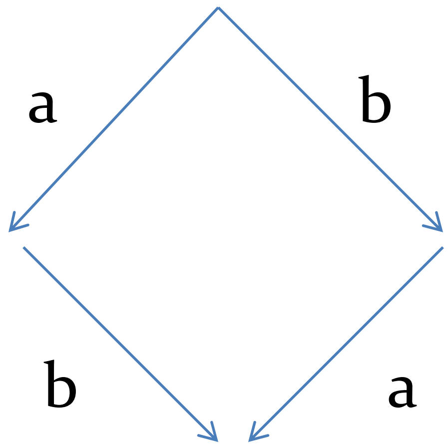
A logic for true concurrency

Paolo Baldan and Silvia Crafa

Universita' di Padova

Models of Concurrency

$$a \mid b \stackrel{?}{=} a.b + b.a$$

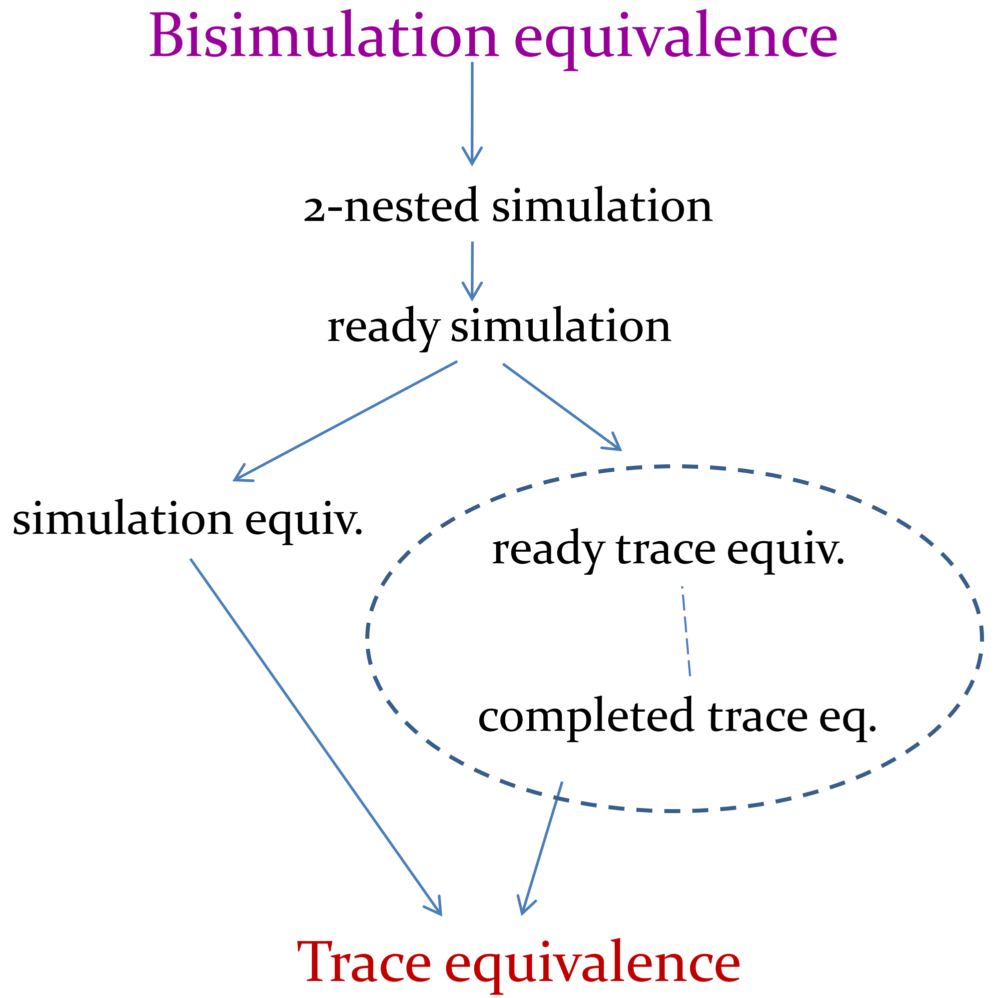


Different ***causal*** properties

Different ***distribution*** properties

The Interleaving world

$a \mid b \sim a.b+b.a$



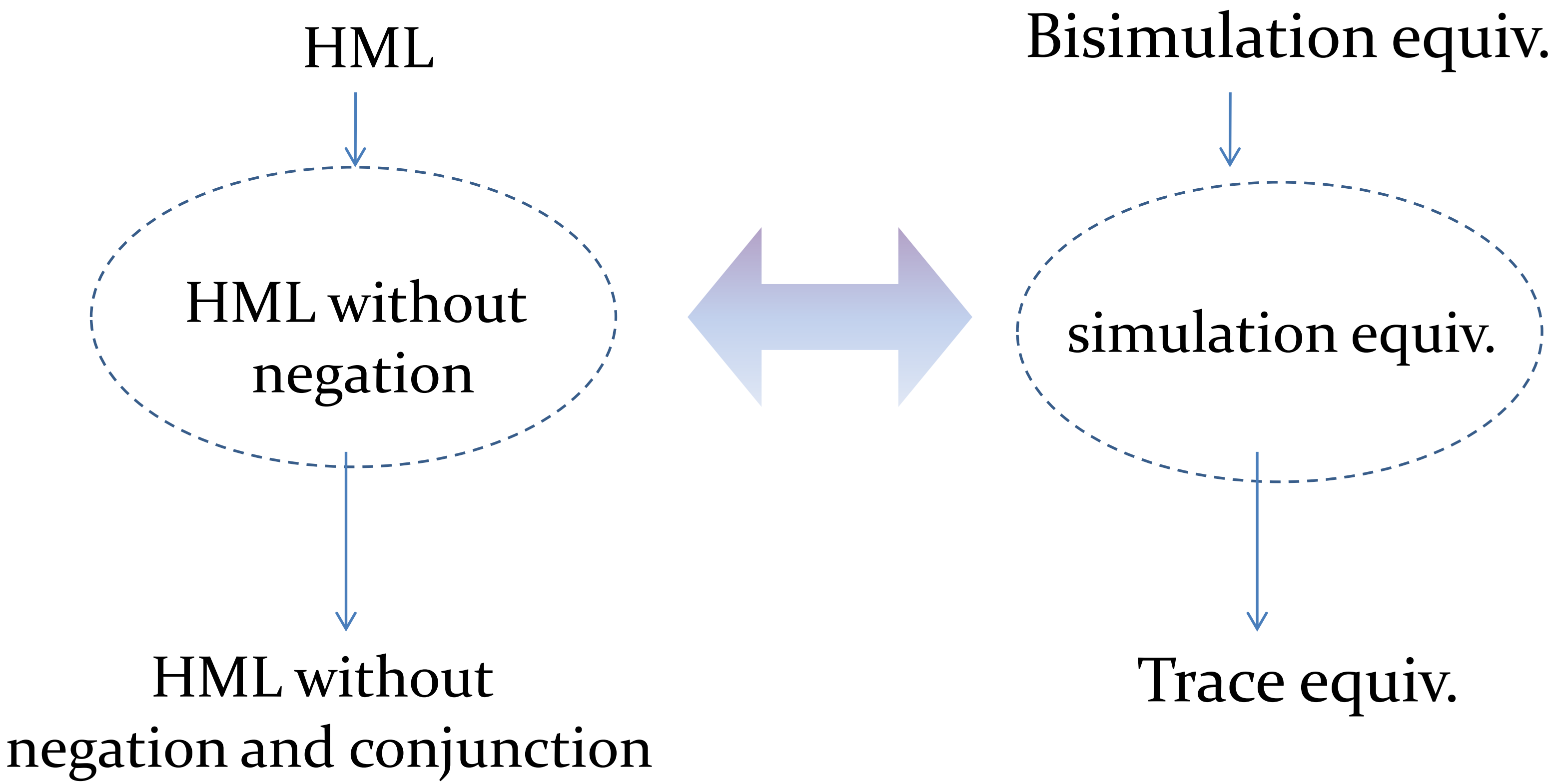
The True-Concurrent world

$a \mid b \neq a.b+b.a$

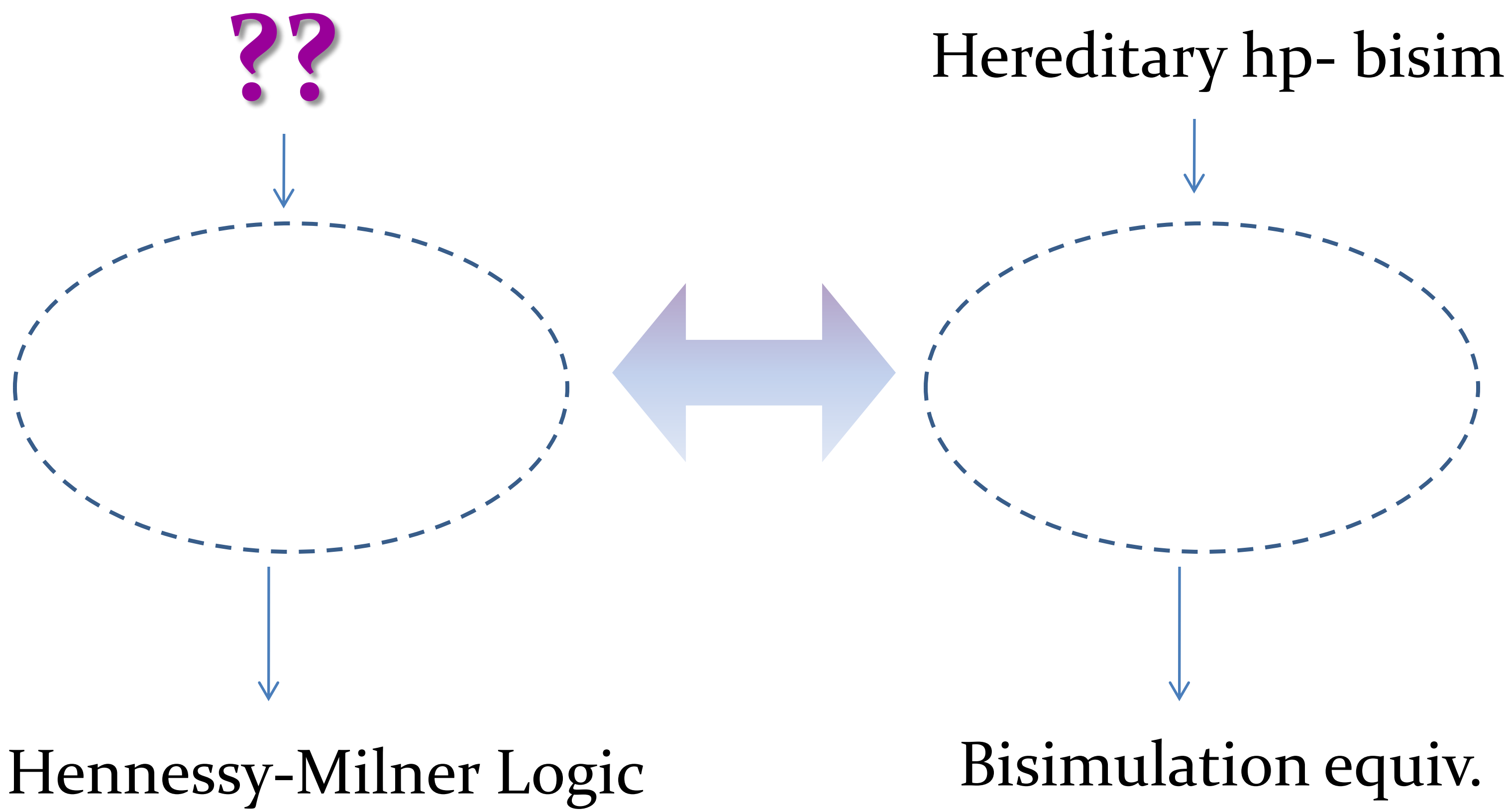
$$
\begin{tikzcd}
 & \text{Hereditary history-preserving bisim} \arrow[d] & \\
 & \text{History-preserving bisim} \arrow[dl] \arrow[dr] & \\
\text{Pomset bisimulation} \arrow[dr] & & \text{weak hp bisim} \arrow[dl] \\
 & \text{Step bisimulation} \arrow[d] & \\
 & \text{(Bisimulation equivalence)} &
\end{tikzcd}
$$

Interleaving world: Logical characterization

Hennessy-Milner Logic $\varphi ::= \top \mid \langle a\rangle\varphi \mid \neg\varphi \mid \varphi \wedge \varphi$



True-concurrent world vs Logic ?



Logics for true-concurrency

[DeNicola-Ferrari 90]

Unique framework for *several* temporal and modal logics.

Captures pomset bisim and

[Hennessy-Stirling 85, Nielsen-Clau

Charaterise hhp-bis with **p**

In absence of autoconcurrency

Different logics for different equivalences!!

[Bradfield-Froschle 02, Gutierrez 09]

Modal logics expressing action independence/causality

Only hp-bisimulation is captured

A single logic for true-concurrency

$\mathcal{L}$		Hereditary hp- bisim
$\downarrow$		$\downarrow$
$\mathcal{L}_{hp}$		hp bisim
$\downarrow$		$\downarrow$
$\mathcal{L}_{p}$	$\Leftrightarrow$	Pomset bisim
$\downarrow$		$\downarrow$
$\mathcal{L}_{s}$		Step bisim
$\downarrow$		$\downarrow$
Hennessy-Milner Logic		Bisimulation equiv.
$\Downarrow$		$\Downarrow$

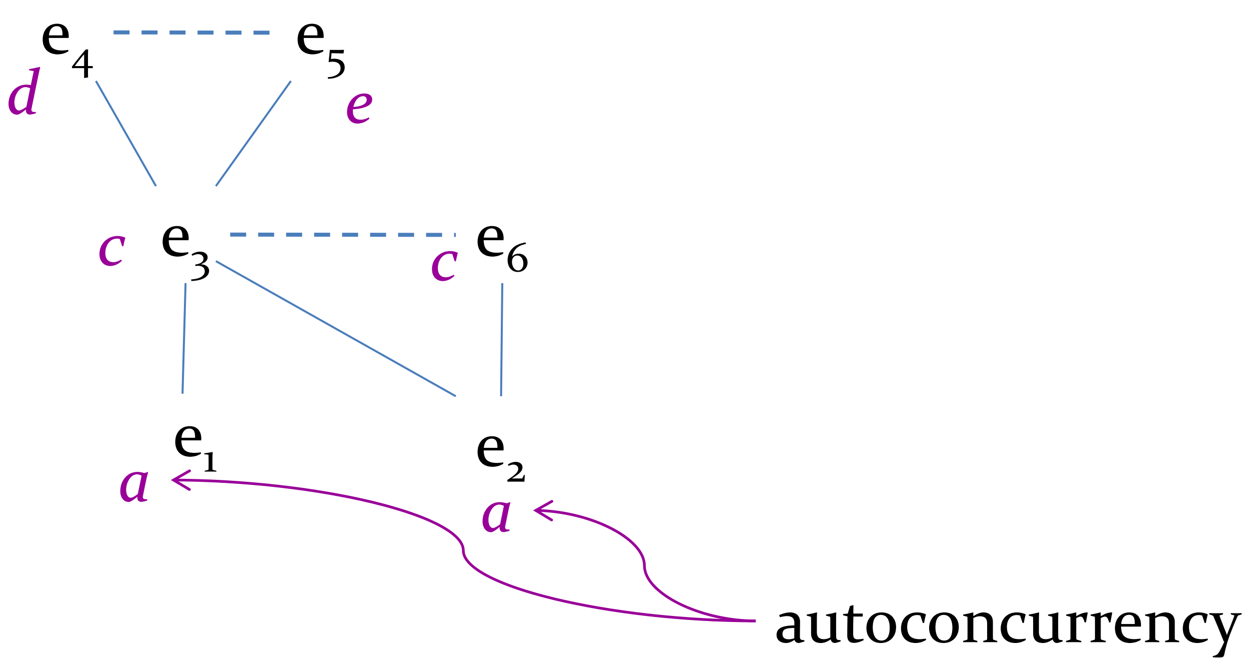
True Concurrent Model: Event Structures

- Computation in terms of events = action occurrence
- A notion of causal dependence between events
- A notion of incompatibility between events
- A labeling to record the action corresponding to the event

$\mathcal{E} = (E, \leq, \#, \lambda)$

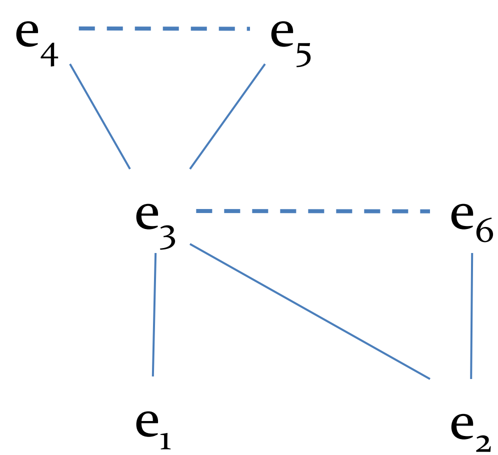
- $\leq$ is a partial order and $\lceil e \rceil = \{e' \mid e' \leq e\}$ is finite
- \# is irreflexive, symmetric and hereditary: if e # e' $\leq$ e'' then e#e''

True Concurrent Model: Event Structures



- e_4 is caused by $\{e_1, e_2, e_3\}$
- (e_1, e_2) and (e_1, e_6) are ***concurrent***
- (e_3, e_6) and (e_5, e_6) are in ***conflict***
- (e_2, e_4) and (e_1, e_6) are ***consistent***

True Concurrent Model: Event Structures



Computation

in terms of

Configurations

causally-closed set of consistent events

$$\emptyset \xrightarrow{e_2} \{e_2\} \xrightarrow{e_6} \{e_2, e_6\}$$

$$\emptyset \xrightarrow{\{e_1, e_2\}} C \xrightarrow{\{e_3, e_5\}} C'$$

step

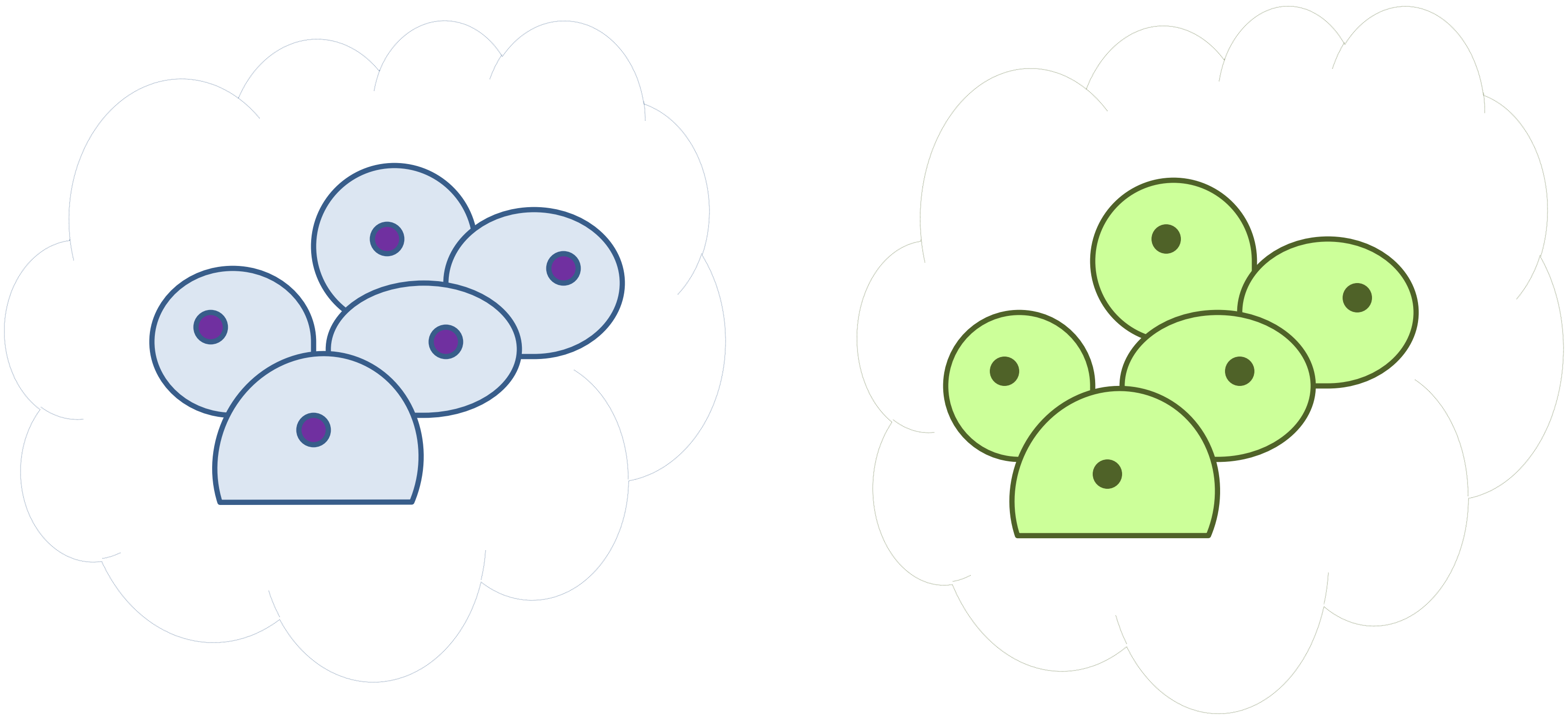
pomset

a run

The True Concurrent Spectrum

$$
\begin{tikzcd}
 & \text{Coherent hhp-bisim} \arrow[d] & \\
 & \text{Hereditary history-preserving bisim} \arrow[d] & \\
 & \text{History-preserving bisimulation} \arrow[dl] \arrow[dr] & \\
\text{Pomset bisimulation} \arrow[dr] & & \text{weak hp bisim} \arrow[dl] \\
 & \text{Step bisimulation} \arrow[d] & \\
 & \text{(Bisimulation equivalence)} &
\end{tikzcd}
$$

Bisimulation Equivalence



A bisimulation is a symmetric relation between configurations s.t. whenever $(C, C') \in R$

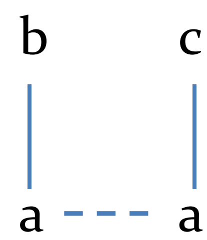
if $C \xrightarrow{e} D$ then $C' \xrightarrow{e'} D'$ with $(D, D') \in R$ and $\lambda(e) = \lambda(e')$

$$\mathcal{E} \sim \mathcal{F} \quad \text{iff} \quad (\emptyset, \emptyset) \in R$$

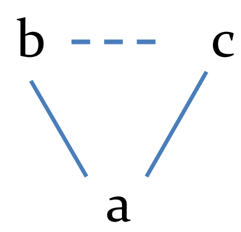
Bisimulation Equivalence

- It captures the branching of a system

a.b +a.c



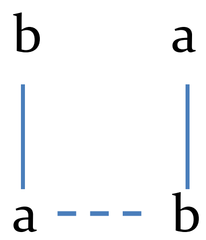
≠



a.(b+c)

- but it cannot observe the concurrency of a system

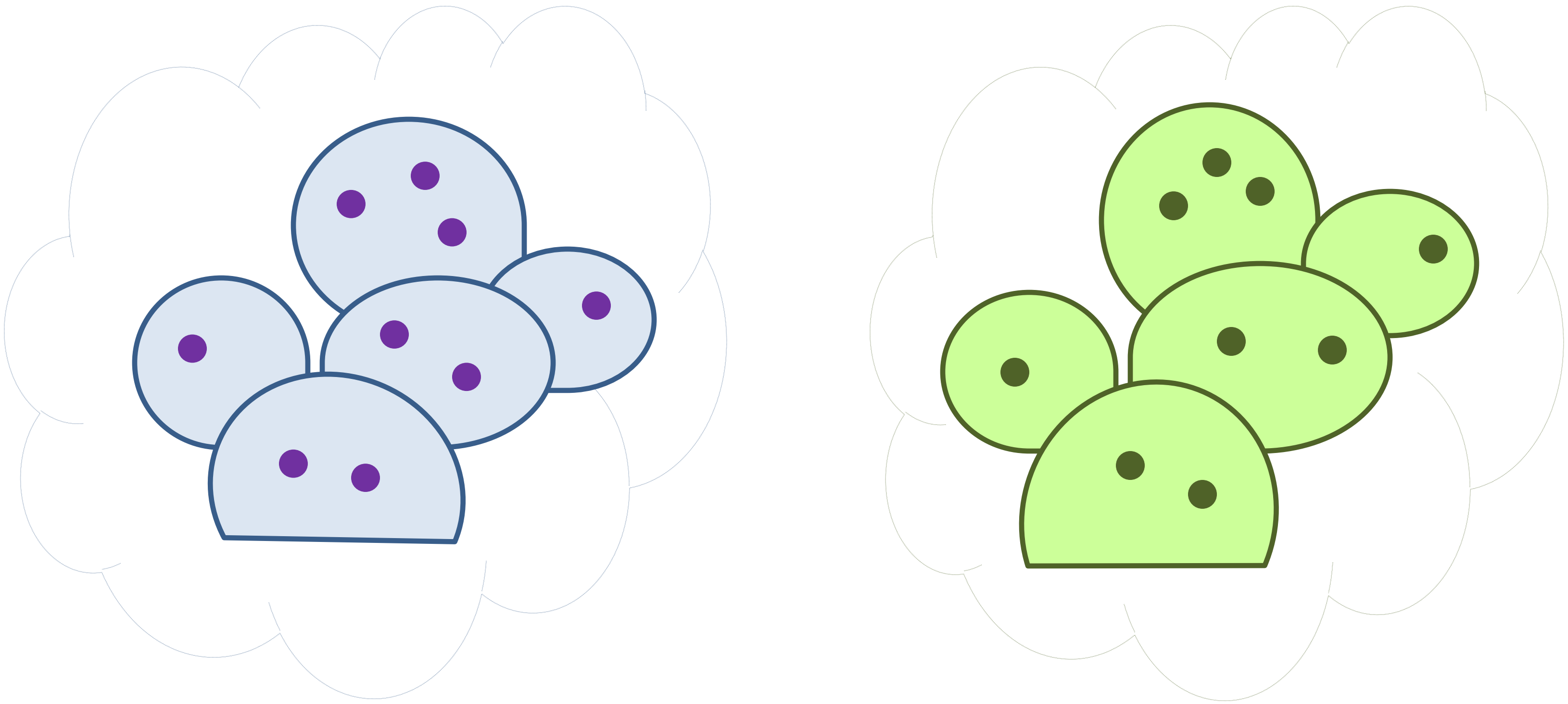
a.b +b.a



=

a | b

Step Bisimulation



whenever $(C, C') \in R$

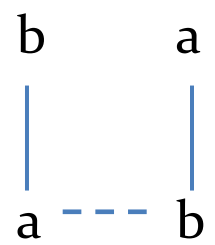
if $C \xrightarrow{X} D$ then $C' \xrightarrow{X'} D'$ with $(D, D') \in R$

and X, X' are isomorphic steps (i.e., sets of concurrent events)

Step Bisimulation

- It captures the concurrency of a system

a.b +b.a



$\neq_s$

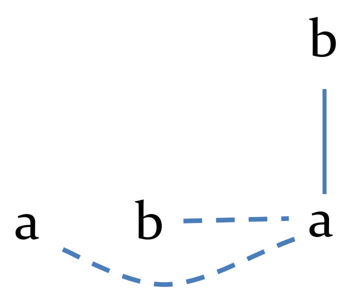
a b

a | b

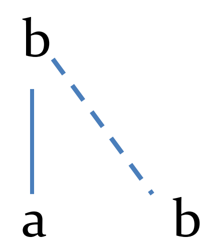
- but it cannot observe the concurrency / causality mix:

$=_s$

a b

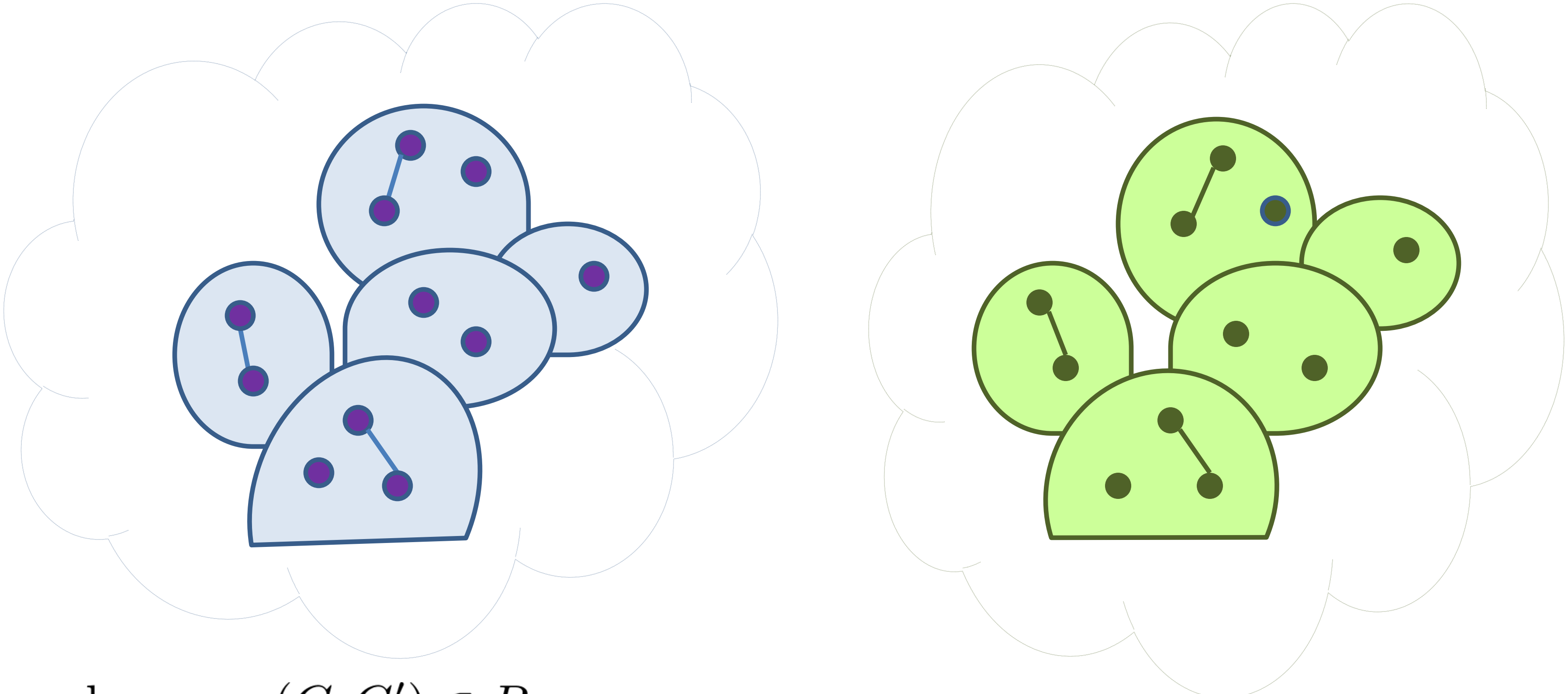


or



There is an occurrence of b causally dependent from a

Pomset Bisimulation



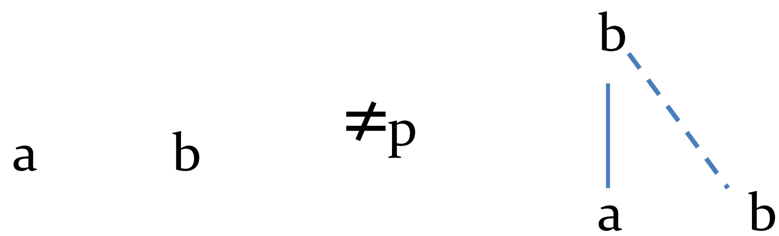
whenever $(C, C') \in R$

if $C \xrightarrow{X} D$ then $C' \xrightarrow{X'} D'$ with $(D, D') \in R$

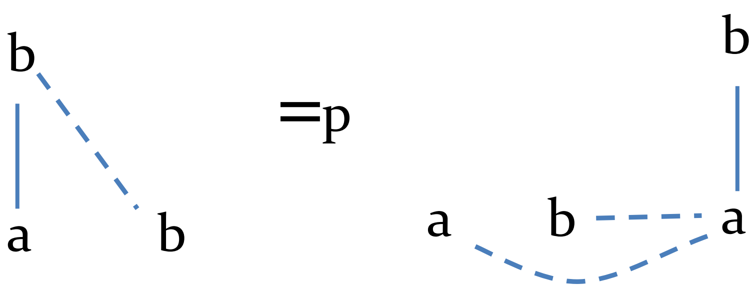
and X, X' are isomorphic pomsets (i.e., p.o. consistent events)

Pomset Bisimulation

- It captures the causality of a system



- but it cannot observe the causality / branching mix:

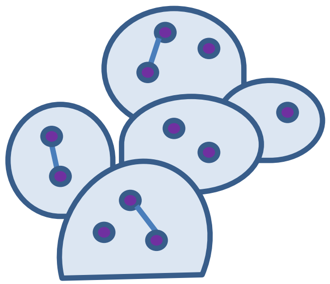


The same pomsets but only in the lhs

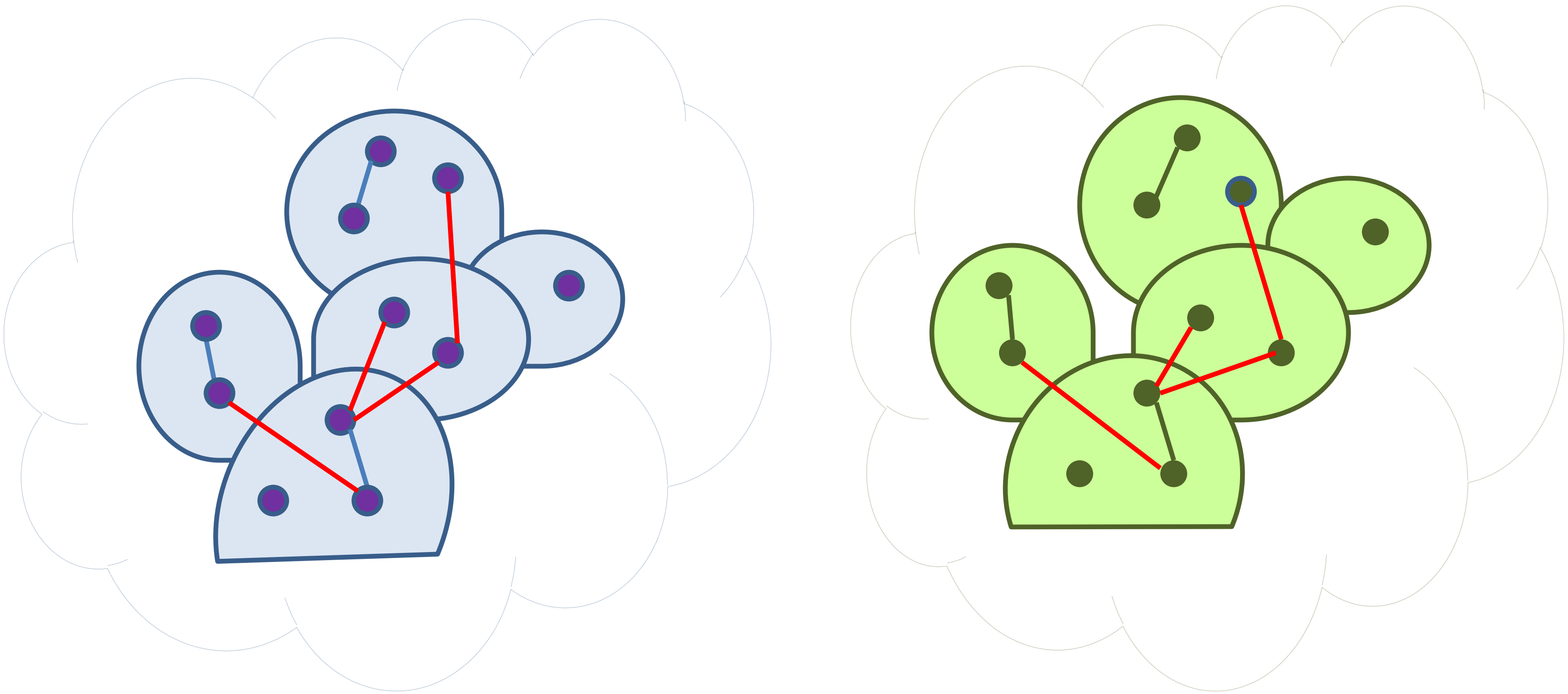
"after a we can choose between a dependent and an independent b"

Pomset Bisimulation

- like bisimulation:
  - it is an interleaving of pomsets (rather than actions)
  - it doesn't observe ***the causal relations between a pomset*** and the next one
- keep the history of already matched transitions
  - Let the two matching runs (entire history of moves) in the game to be pomset-isomorphic
  - ***let the <u>history grow pomset-isomorphically</u>***



History-preserving Bisimulation



whenever $(C, f, C') \in R$

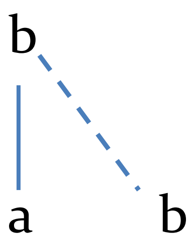
if $C \xrightarrow{e} D$ then $C' \xrightarrow{e'} D'$ with $(D, f[e \to e'], D') \in R$

where $f[e \to e']$ is a label-preserving iso extending f

History-preserving Bisimulation

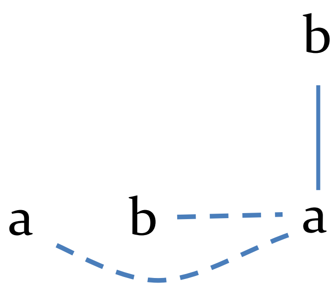
- It captures the causality / branching interplay

"causal equivalence"



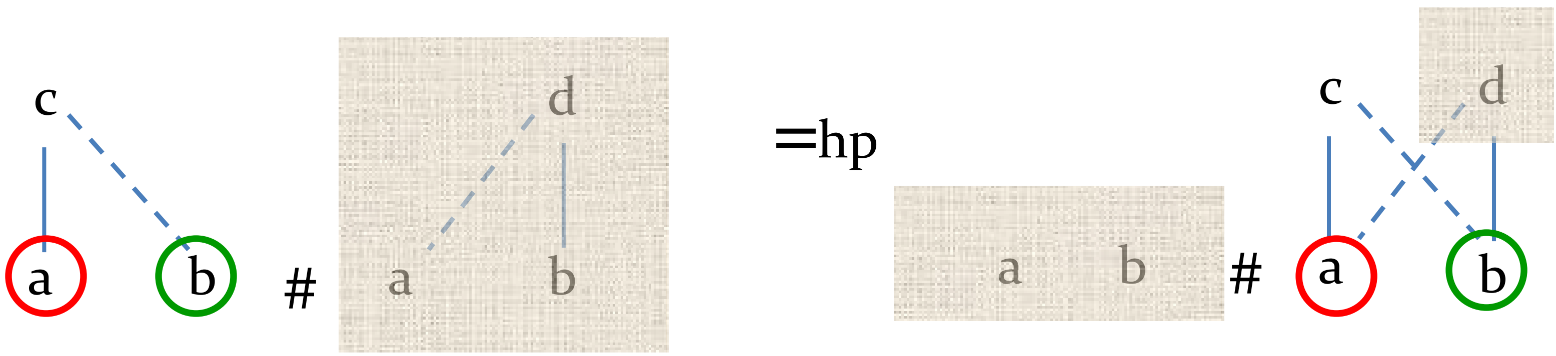
$=_p$

$\neq_{hp}$



- It does not capture the interplay between causality – concurrency - branching

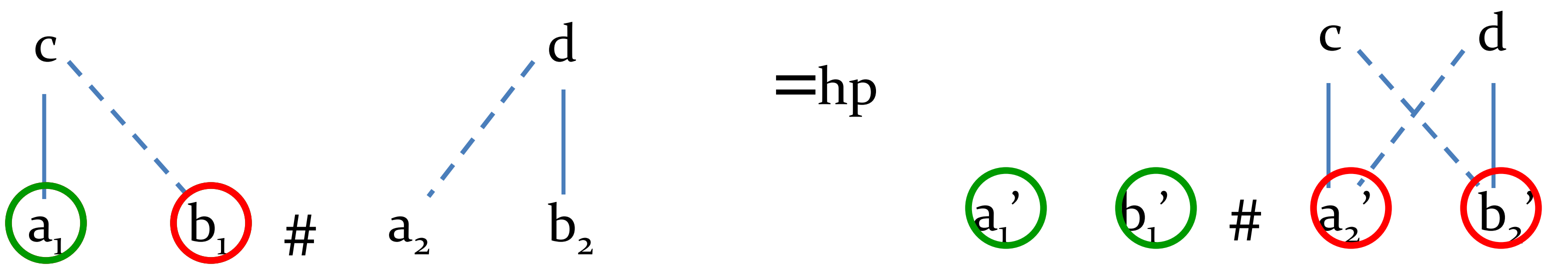
History-preserving Bisimulation



And similarly the other way round

- *c* and *d* depend on conflicting vs. concurrent *a* and *b* !!
  - Hp-bisim hides such a difference:
    - the ***execution*** of an event ***rules out any conflicting*** event
    - there is the same causality

History-preserving Bisimulation



a_1, b_1 can be matched in principle either by a_1', b_1' or a_2', b_2'

- the ***choice depends on the order in which they are linearized***

  (a_1, b_1 are concurrent)

- a_1, b_1 are independent, but the execution of one affects the "behavioral environment"/ the future of the other

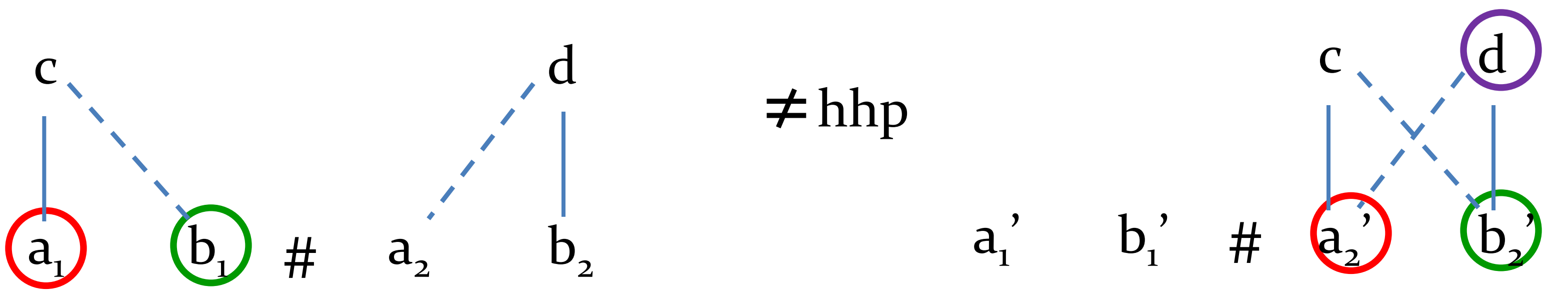
How can we formalize this difference?

Hereditary History-preserving Bisimulation

whenever $(C, f, C') \in R$

- if $C \xrightarrow{e} D$ then $C' \xrightarrow{e'} D'$ with $(D, f[e \to e'], D') \in R$
- if $D \xrightarrow{e} C$ then $D' \xrightarrow{e'} C'$ with $(D, f|_D, D') \in R$

Backward moves!!



$$\xrightarrow{a1} \quad \xrightarrow{b1} \quad \xleftarrow{a1} \quad \not\xrightarrow{d}$$

$$\xrightarrow{a2'} \quad \xrightarrow{b2'} \quad \xleftarrow{a2'} \quad \xrightarrow{d}$$

Hereditary History-preserving Bisimulation

a|(b+c) + a|b + b|(a+c) a|(b+c) + b|(a+c)

a b c # a b # a c b

a b c # a c b

- no causality
- In lhs only: a couple of independent (a,b) so that none can appear in parallel with c
- Hp-bisim hides such a difference
- a and b are independent but their linearization affects the behavioral environment
- The backtracking distinguishes them

Hereditary History-preserving Bisimulation

The backtracking can distinguish them

$a|(b+c)\ + a|b + b|(a+c)$ $\qquad\qquad$ $a|(b+c) + b|(a+c)$

a $\quad$ b $\quad$ c $\quad$ # $\quad$ a $\quad$ b $\quad$ # $\quad$ a $\quad$ c $\quad$ b

$a_1 \quad b_1 \quad c_1 \quad \# \quad a_2 \quad c_2 \quad b_2$

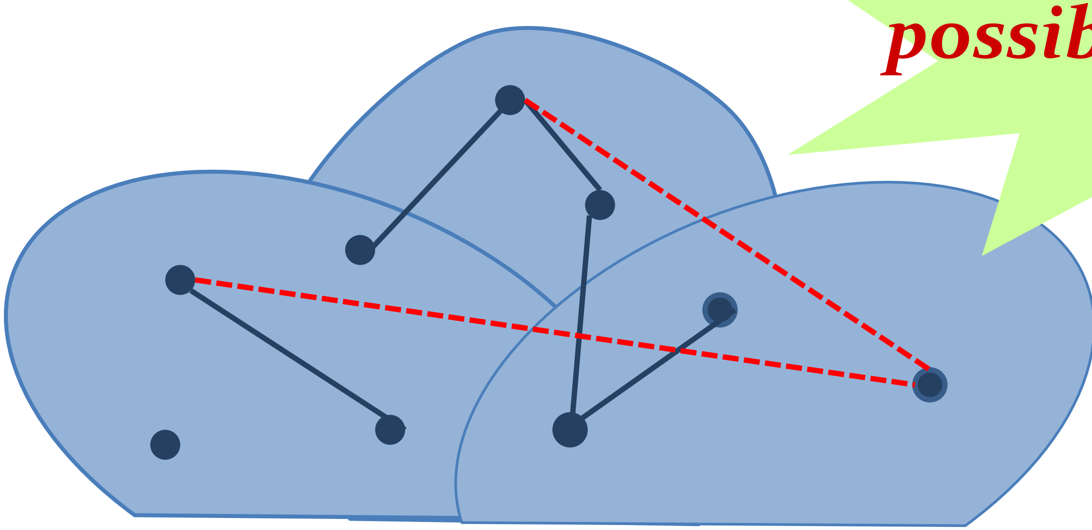
$$\xrightarrow{a} \quad \xrightarrow{b} \quad \xleftarrow{a} \quad \not\xrightarrow{c}$$

$$\xrightarrow{a2} \quad \xrightarrow{b2} \quad \xleftarrow{a2} \quad \xrightarrow{c}$$

Hereditary History-preserving Bisimulation

What kind of ***forward observation*** does backtracking correspond to?

alternative,
possibly conflicting
futures



$$\emptyset \xrightarrow{X_1} C_1 \xleftarrow{Y_1} C_2 \xrightarrow{X_2} C_3 \xleftarrow{Y_2} C_4 \xrightarrow{X_3} C_5$$

A logic for true concurrency

Var : denumerable set of variables ranged over by *x, y, z*, ...

$$\varphi \;::=\; (\mathbf{x}, \overline{\mathbf{y}} < \mathsf{a}\, z)\,\varphi \;\mid\; \langle z \rangle\,\varphi \;\mid\; \varphi \wedge \varphi \;\mid\; \neg\varphi \;\mid\; \top$$

Interpreted over event structures:
a formula is evaluated in a configuration *C*, with an environment $\eta : Var \to E$

$$C \models_\eta \varphi$$

the **current state** of the computation

records the events bound to variables

describes a set of **possible futures for *C***

A logic for true concurrency

$$\varphi \;::=\; (\mathbf{x}, \overline{\mathbf{y}} < \mathsf{a}\, z)\, \varphi \;\mid\; \langle z \rangle\, \varphi \mid\; \varphi \wedge \varphi \;\mid\; \neg \varphi \;\mid\; \top$$

Event-based logic

it binds z to e so that it can be later referred to in φ

$$C \models_\eta (\mathbf{x}, \overline{\mathbf{y}} < \mathsf{a}\, z)\, \varphi$$

declares the existence of an event *e* in the future of *C* s.t.

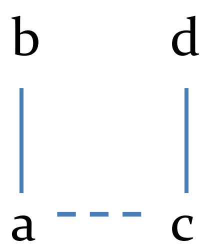
$$\eta(\mathbf{x}) < e,\;\; \eta(\mathbf{y}) || e,\;\; \lambda(e) = \mathsf{a} \text{ and } C \models_{\eta[z \to e]} \varphi$$

$$C \models_\eta \langle z \rangle\, \varphi$$

the event $\eta(z)$ can be executed form *C*, leading to *C'* s.t.

$$C' \models_\eta \varphi$$

A logic for true concurrency



$$\emptyset \models_{\emptyset} (\mathsf{b}\,x)\top$$

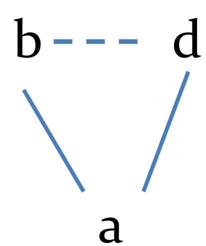
there is a future evolution that enables **b**

$$\emptyset \models_{\emptyset} (\mathsf{b}\,x)\top \wedge (\mathsf{d}\,y)\top$$

there are two (incompatible) futures

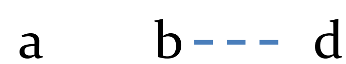
$$\emptyset \not\models_{\emptyset} (\mathsf{a}\,z)\langle z\rangle\,((\mathsf{b}\,x) \wedge (\mathsf{d}\,y))$$

executing **a** disallows the future containing **d**



$$\emptyset \models_{\emptyset} (\mathsf{a}\,z)\langle z\rangle\,((\mathsf{b}\,x) \wedge (\mathsf{d}\,y))$$

$$\emptyset \not\models_{\emptyset} (\mathsf{a}\,z)\langle z\rangle\,(\overline{z} < \mathsf{b}\,x)$$



$$\emptyset \models_{\emptyset} (\mathsf{a}\,z)\langle z\rangle\,((\mathsf{b}\,x) \wedge (\mathsf{d}\,y))$$

$$\emptyset \models_{\emptyset} (\mathsf{a}\,z)\langle z\rangle\,(\overline{z} < \mathsf{b}\,x)$$

Examples and notation

- Immediate execution

$$\langle\!|\mathbf{x}, \overline{\mathbf{y}} < \mathsf{a}\ z|\!\rangle\ \varphi$$

stands for $(\mathbf{x}, \overline{\mathbf{y}} < \mathsf{a}\ z)\langle z\rangle\ \varphi$ that chooses an event and immediately executes it

- Step

$$((\mathbf{x}, \overline{\mathbf{y}} < \mathsf{a}\ z)\ \otimes\ (\mathbf{x}', \overline{\mathbf{y}'} < \mathsf{b}\ z'))\ \varphi$$

stands for $((\mathbf{x}, \overline{\mathbf{y}} < \mathsf{a}\ z)\ (\mathbf{x}', \overline{\mathbf{y}', \mathbf{z}} < \mathsf{b}\ z'))\ \varphi$ which declares the existence of two concurrent events

Examples and notation

- Immediate execution

$$\underline{((\mathsf{a}\,x) \otimes (\mathsf{b}\,y))}\,(\underline{(x < \mathsf{c}) \otimes (y < \mathsf{d})})\ \top$$

sta

im

$$(\llparenthesis\, \mathsf{a} \,\rrparenthesis \otimes \llparenthesis\, \mathsf{b} \,\rrparenthesis \otimes \llparenthesis\, \mathsf{c} \,\rrparenthesis)\ \varphi$$

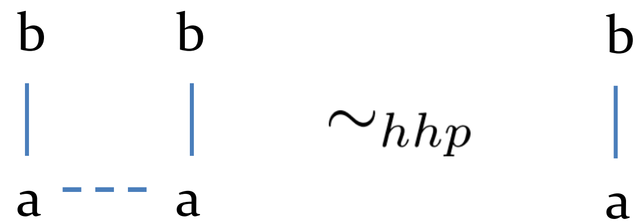
- Step

$$(\underline{\llparenthesis \mathsf{a}\,x \rrparenthesis \otimes \llparenthesis \mathsf{a}\,y \rrparenthesis})\,(\underline{\llparenthesis x < \mathsf{b} \rrparenthesis \otimes \llparenthesis \overline{y} < \mathsf{b} \rrparenthesis})\,\varphi$$

stands for $((\mathbf{x}, \overline{\mathbf{y}} < \mathsf{a}\,z)\,(\mathbf{x}', \overline{\mathbf{y}', \mathbf{z}} < \mathsf{b}\,z'))\ \varphi$ which declares the existence of two concurrent events

Well-formedness

The full logic is too powerful: it also observe conflicts!



$$\mathcal{E}_1 \models, \mathcal{E}_2 \not\models \ (\mathsf{a}\, x)(\mathsf{b}\, y)\langle x\rangle\neg\langle y\rangle$$

Well-formedness syntactically ensures that

- free variables in any subformula will always refer to events consistent with the current config.
- the variables used as causes/non causes in quantifications will be bound to consistent events

Logical Equivalence

- An e.s. satisfies a *closed* formula φ: $\mathcal{E} \models \varphi$ when $\mathcal{E}, \emptyset \models_{\emptyset} \varphi$

- Two e.s. are **logically equivalent in the logic** $\mathcal{L}$:

$$\mathcal{E}_1 \equiv_{\mathcal{L}} \mathcal{E}_2 \quad \text{when} \quad \mathcal{E}_1 \models \varphi \ \text{ iff } \ \mathcal{E}_2 \models \varphi$$

Theorem: $\mathcal{E}_1 \equiv_{\mathcal{L}} \mathcal{E}_2 \quad \text{iff} \quad \mathcal{E}_1 \sim_{hhp} \mathcal{E}_2$

The logical equivalence induced by the full logic is hhp-bisimilarity

A single logic for true-concurrency

no back moves

$\mathcal{L}$		Hereditary hp- bisim
$\downarrow$		$\downarrow$
$\mathcal{L}_{hp}$		hp bisim
$\downarrow$		$\downarrow$
$\mathcal{L}_{p}$	$\Leftrightarrow$	Pomset bisim
$\downarrow$		$\downarrow$
$\mathcal{L}_{s}$		Step bisim
$\downarrow$		$\downarrow$
Hennessy-Milner Logic		Bisimulation equiv.
$\Downarrow$		$\Downarrow$

Logical Spectrum: HM Logic

Hennessy-Milner logic corresponds to the fragment $\mathcal{L}_{\mathrm{HM}}$:

$$\varphi \ ::= \ \langle\!|\, \mathsf{a}\ x \,|\!\rangle\, \varphi \mid \varphi \wedge \varphi \mid \neg\varphi \mid \top$$

- No references to causally dependent/concurrent events
- Whenever we state the existence of an event, we must execute it

Theorem: $\mathcal{E}_1 \equiv_{\mathcal{L}_{HM}} \mathcal{E}_2 \quad \text{iff} \quad \mathcal{E}_1 \sim \mathcal{E}_2$

The logical equivalence induced by $\mathcal{L}_{\mathrm{HM}}$ is bisimilarity

Logical Spectrum: Step Logic

The fragment $\mathcal{L}_s$:

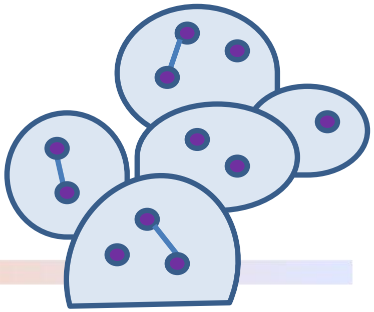
$$\varphi \ ::= \ (\langle\!|\mathsf{a}_1\, x_1|\!\rangle \otimes \cdots \otimes \langle\!|\mathsf{a}_n\, x_n|\!\rangle)\,\varphi \mid \varphi \wedge \varphi \mid \neg\varphi \mid \top$$

- Predicates on the possibility of performing a parallel step
- No references to causally dependent/concurrent events between steps
- Generalizes $\mathcal{L}_{\mathrm{HM}}$

Theorem: $$\mathcal{E}_1 \equiv_{\mathcal{L}_s} \mathcal{E}_2 \quad \text{iff} \quad \mathcal{E}_1 \sim_s \mathcal{E}_2$$

The logical equivalence induced by $\mathcal{L}_s$ is step bisimulation

Logical Spectrum: Pomset Logic



The fragment $\mathcal{L}_{\mathrm{p}}$:

$$\varphi \ ::= \ \langle\!|\mathbf{x}, \overline{\mathbf{y}} < \mathsf{a}\, z |\!\rangle\, \varphi \mid \varphi \wedge \varphi \mid \neg \varphi \mid \top$$

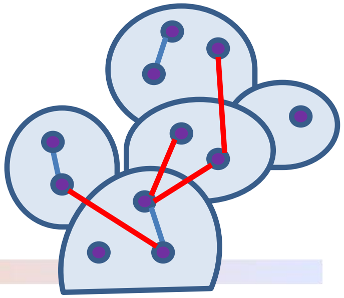
where $\neg$, $\wedge$ are used only **on closed formulae**

- Predicates on the possibility of executing a pomset transition
- Closed formula $\leftrightarrow$ execution of a pomset
- Causal links only within a pomset but not between different pomsets

Theorem: $\mathcal{E}_1 \equiv_{\mathcal{L}_p} \mathcal{E}_2 \quad \text{iff} \quad \mathcal{E}_1 \sim_p \mathcal{E}_2$

The logical equivalence induced by $\mathcal{L}_{\mathrm{p}}$ is pomset bisimulation

Logical Spectrum: History Preserving Logic



The fragment $\mathcal{L}_{\mathrm{hp}}$:

$$\varphi \;::=\; \langle\!|\mathbf{x}, \overline{\mathbf{y}} < \mathsf{a}\; z|\!\rangle\, \varphi \;\mid\; \varphi \wedge \varphi \;\mid\; \neg\varphi \;\mid\; \top$$

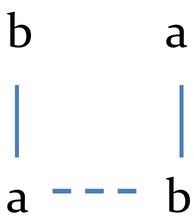
- Besides pomset execution, it also predicates about its dependencies with previously executed events

- **quantify + execute $\rightarrow$ no quantification over conflicting events**

Theorem: $\mathcal{E}_1 \equiv_{\mathcal{L}_{hp}} \mathcal{E}_2 \quad \text{iff} \quad \mathcal{E}_1 \sim_{hp} \mathcal{E}_2$

The logical equivalence induced by $\mathcal{L}_{\mathrm{hp}}$ is hp-bisimulation

Logical Spectrum: Separation Examples



$\sim$

$\not\sim_s$

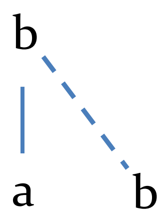
a b

$$\mathcal{E}_1 \not\models, \mathcal{E}_2 \models \quad \langle\!| \mathsf{a} |\!\rangle \otimes \langle\!| \mathsf{b} |\!\rangle \quad \in \mathcal{L}_s$$

a b

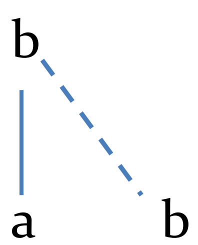
$\sim_s$

$\not\sim_p$



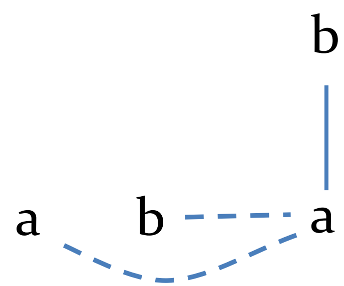
$$\mathcal{E}_1 \not\models, \mathcal{E}_2 \models \quad \langle\!| \mathsf{a}\; x |\!\rangle \langle\!| x < \mathsf{b}\; y |\!\rangle \quad \in \mathcal{L}_p$$

Logical Spectrum: Separation Examples



$\sim_p$

$\not\sim_{hp}$

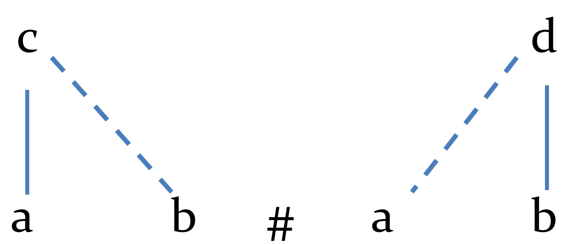


The same pomsets but only in the lhs

“after a we can choose between a dependent and an independent b”

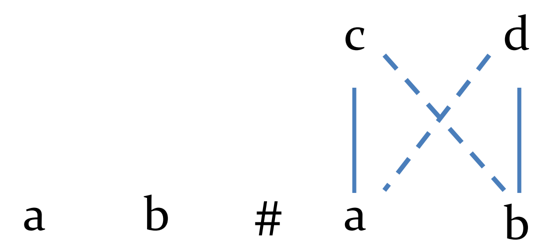
$$\mathcal{E}_1 \models, \mathcal{E}_2 \not\models \quad \langle\!|\mathsf{a}\,x|\!\rangle\,(\langle\!|x < \mathsf{b}\,y|\!\rangle \wedge \langle\!|\overline{x} < \mathsf{b}\,z|\!\rangle) \qquad \in \mathcal{L}_{hp}$$

Logical Spectrum: Separation Examples



$\sim_{hp}$

$\not\sim_{hhp}$



The same causality but

c and d depend on conflicting vs. concurrent a and b

$$\mathcal{E}_1 \not\models, \mathcal{E}_2 \models \quad ((\mathsf{a}\,x) \otimes (\mathsf{b}\,y))\,((x < \mathsf{c}\,z) \wedge (y < \mathsf{d}\,z')) \qquad \in \mathcal{L}_{hhp}$$

observe without executing:

conflicting futures

$$\mathcal{E}_1 \not\models, \mathcal{E}_2 \not\models \quad (\langle\!|\mathsf{a}\,x|\!\rangle \otimes \langle\!|\mathsf{b}\,y|\!\rangle)\,((x < \mathsf{c}\,z) \wedge (y < \mathsf{d}\,z')) \qquad \in \mathcal{L}_{hp}$$ **!!**

Logical Spectrum: Separation Examples

a|(b+c) + a|b + b|(a+c) $\sim_{hp}$ a|(b+c) + b|(a+c)

$\not\sim_{hhp}$

The same causality but in lhs only

a couple of independent (a,b) so that

none can appear in parallel with c

$$\mathcal{E}_1 \models, \mathcal{E}_2 \not\models \quad ((\mathsf{a}\, x) \otimes (\mathsf{b}\, y))\,(\neg(\overline{x} < \mathsf{c}\, z) \wedge \neg(\overline{y} < \mathsf{c} z')) \qquad \in \mathcal{L}_{hhp}$$

observe without executing:

Future work

Different equivalences in a simple, unitary logical framework

- ***Study the operational spectrum:***
  - observe without executing, but only predicate on consistent futures lies in between hp and hhp-bis.
  - hp is decidable and hhp is undecidable even for finite state systems. Characterise decidable equiv.

- ***Study the logical spectrum:***
  - encode other logics in L
  - add recursion to express properties like

    any a-action can be always followed by a causally related b-action

    an a-action can be always executed in parallel with a b-action

- ***Verification:*** model checking, auotmata, games,...