A logic for true concurrency

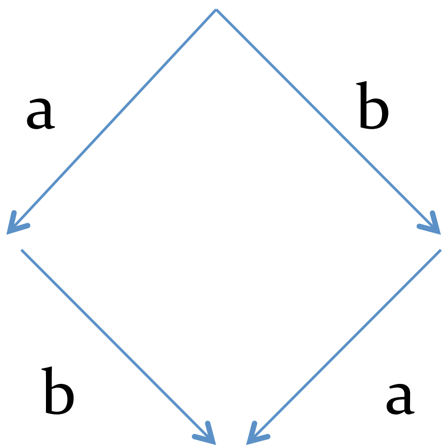
Paolo Baldan

Joint work with Silvia Crafa

University of Padova

Interleaving vs. True concurrency

$$a \mid b \overset{?}{\sim} a.b + b.a$$

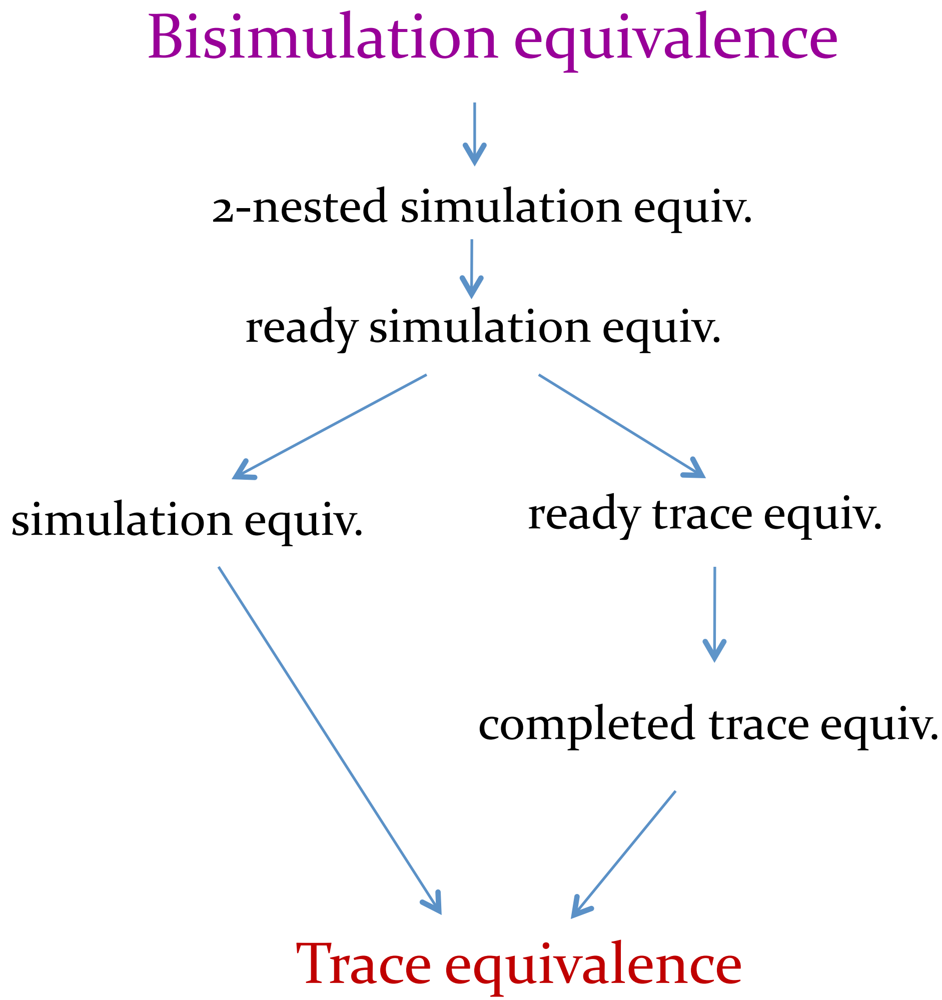


Different ***causal*** properties

Different ***distribution*** properties

Interleaving world

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



Interleaving world: Logical characterization

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

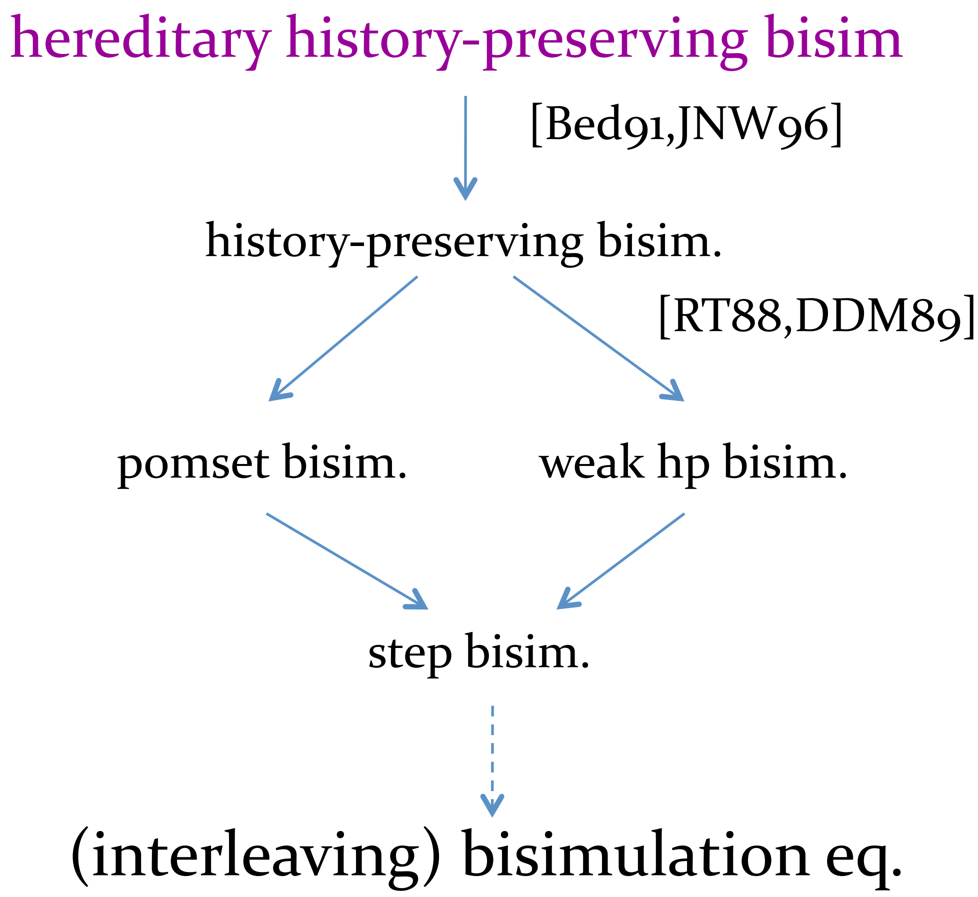
HML ↓ HML without negation ↓ HML without negation and conjunction

⟺

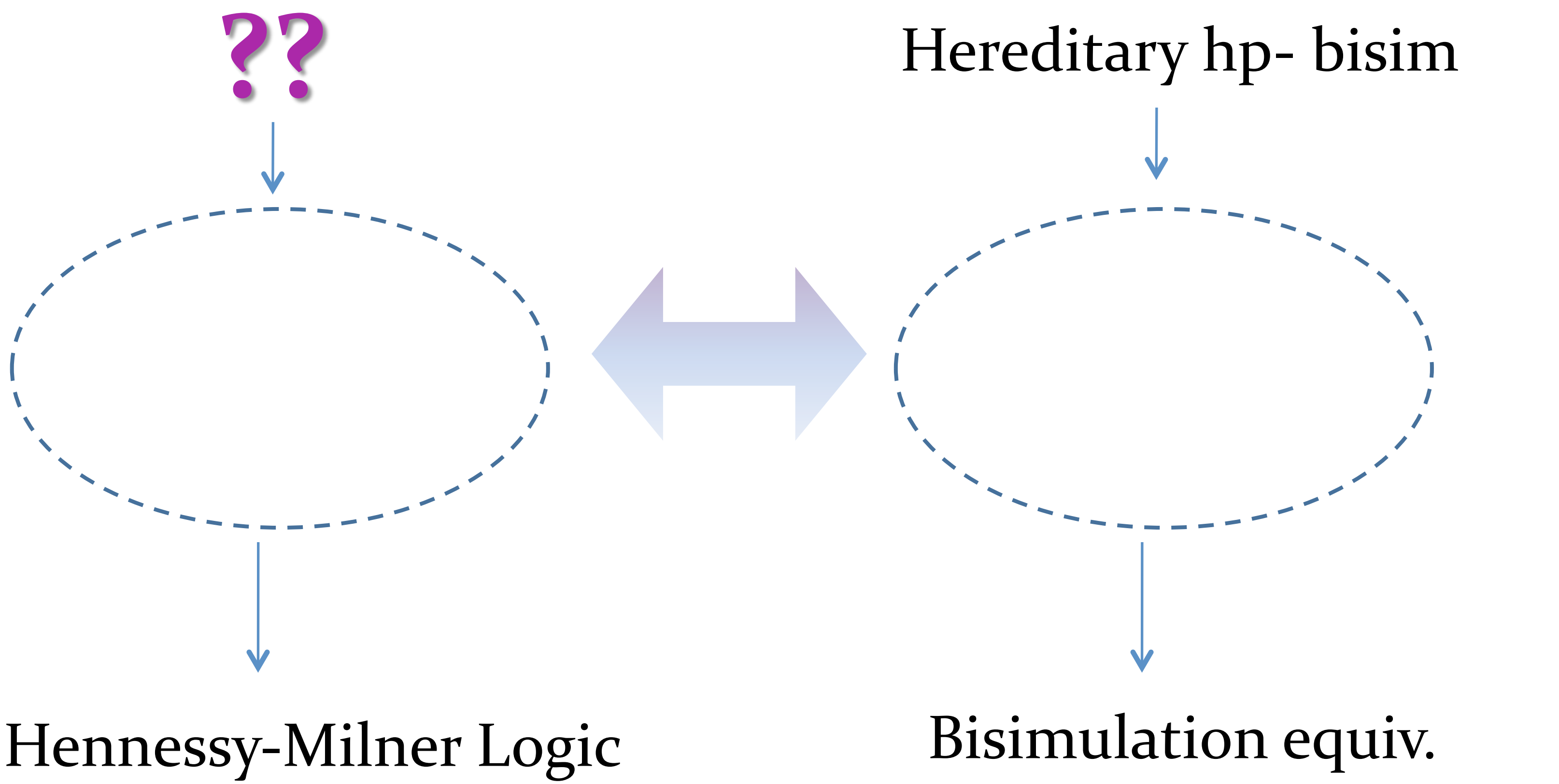
bisimulation equiv. ↓ simulation equiv. ↓ trace equiv.

True-Concurrent world

$a \mid b \not\sim a.b + b.a$



True-concurrent world vs Logic ?



Logics for true-concurrency

[DeNicola-Ferrari 90]

Framework for *several* temporal logics.

Pomset bisim and weak hp-bi

[Hennessy-Stirling 85, Nielsen-Clau

Charaterise hhp-bis with **past**

In absence of autoconcurrency

[Bradfield-Froschle 02, Gutierrez 09]

Modal logics expressing action independence/causality

Captures hp-bisimulation

Different logics for different equivalences!!

Our Proposal

- A logic for true concurrency which allow to predicate on
  - events
  - their dependencies

~ independence friendly modal logic [Bradfield]

A single logic for true-concurrency

$\mathcal{L}$		hereditary hp bisim
↓		↓
$\mathcal{L}_{hp}$		hp bisim
↓		↓
$\mathcal{L}_{p}$	⟺	pomset bisim
↓		↓
$\mathcal{L}_{s}$		step bisim
↓		↓
Hennessy-Milner Logic		bisimulation equiv.
⇓		⇓

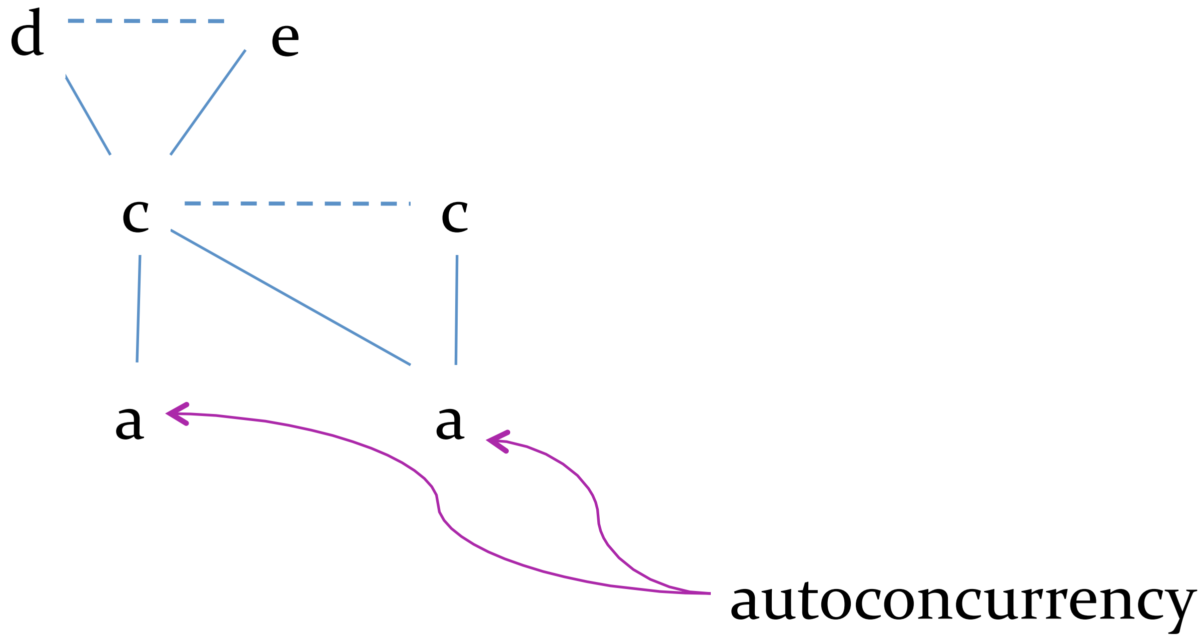
True Concurrent Model: Event Structures

- Computation in terms of events = action occurrence
- Causality / incompatibility between events
- A labeling to record the actions corresponding to events

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

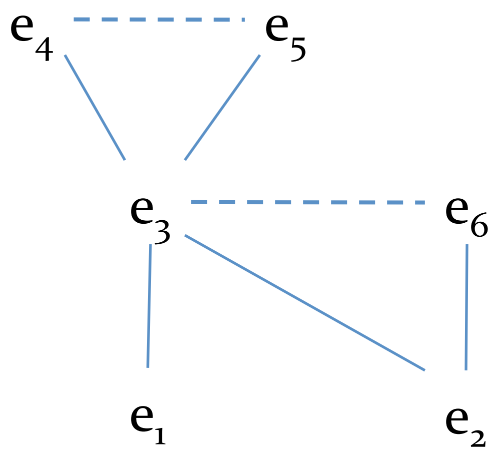
- $\leq$ is a partial order and $\lceil e \rceil = \{e' \mid e' \leq e\}$ is finite
- # is irreflexive, symmetric and hereditary: if e # e' $\leq$ e'' then e#e''

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

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

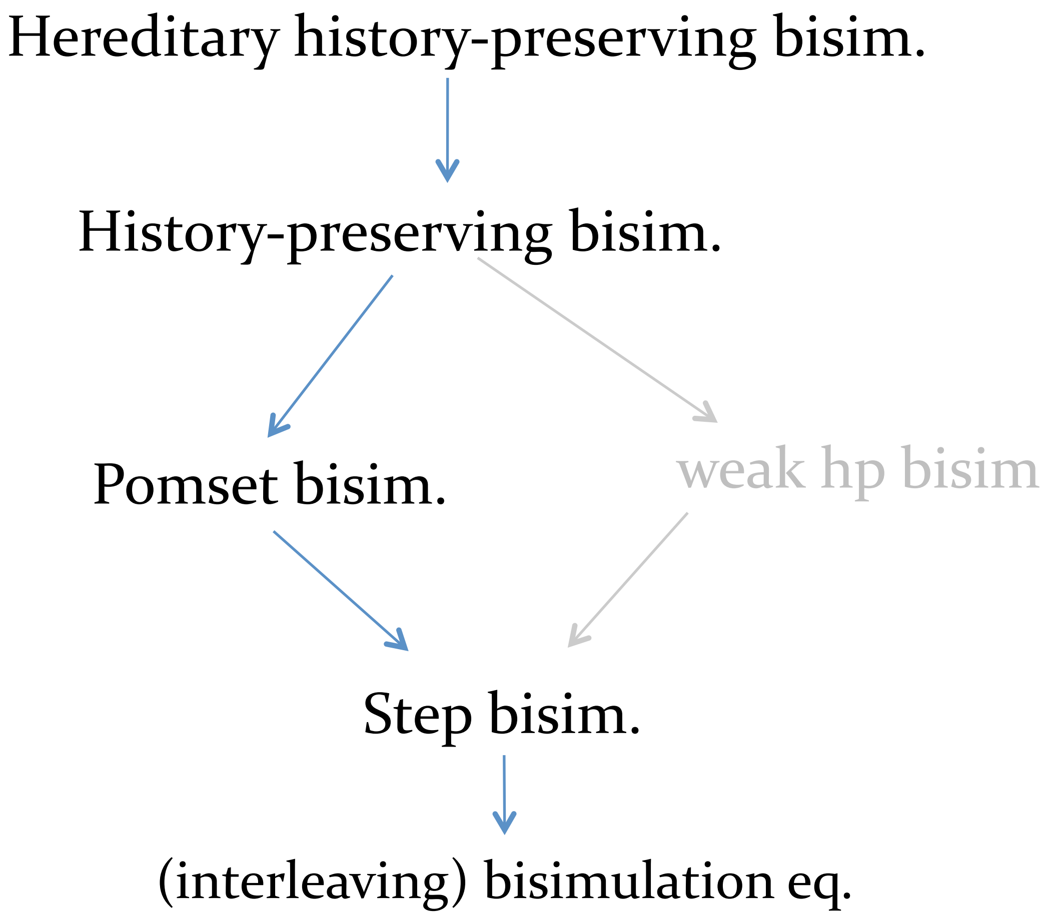
$$\emptyset \xrightarrow{\{e_1,e_2\}} C \xrightarrow{\{e_3,e_5\}} C'$$

step

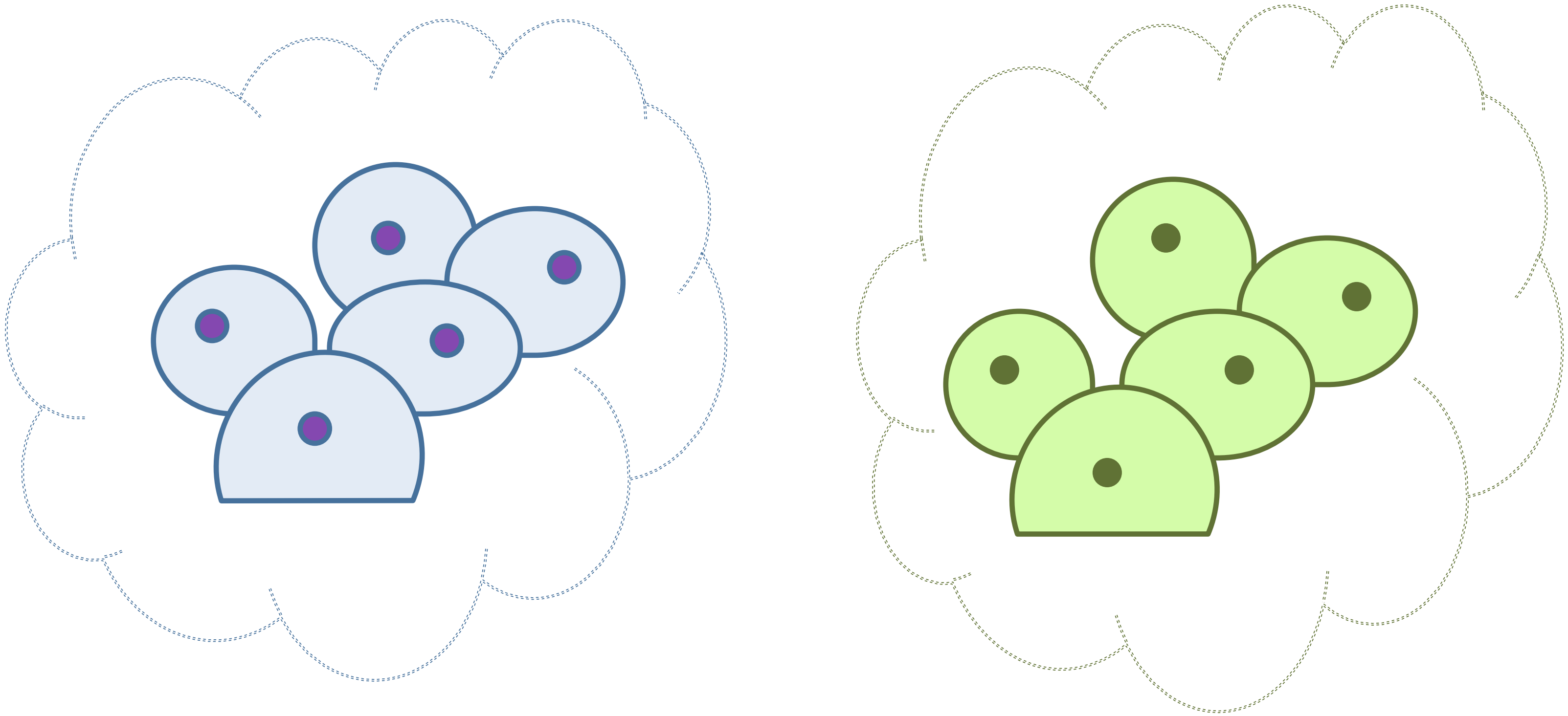
pomset

a run

True Concurrent Spectrum



(Interleaving) Bisimulation



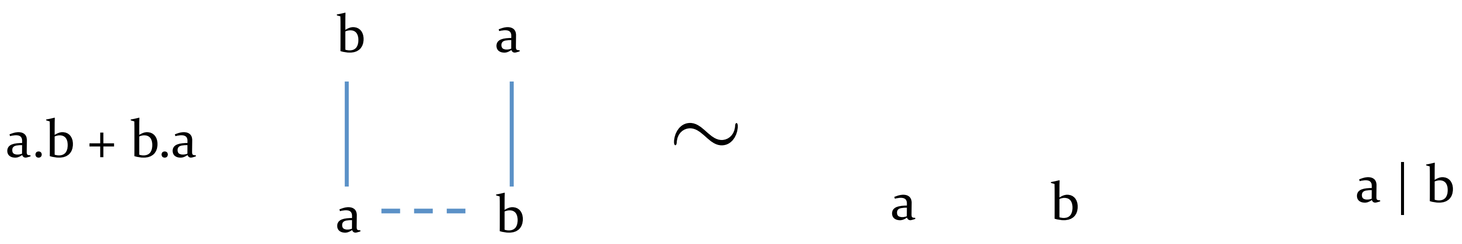
A bisimulation is a symmetric relation between configurations s.t. whenever $(C, C') \in R$

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

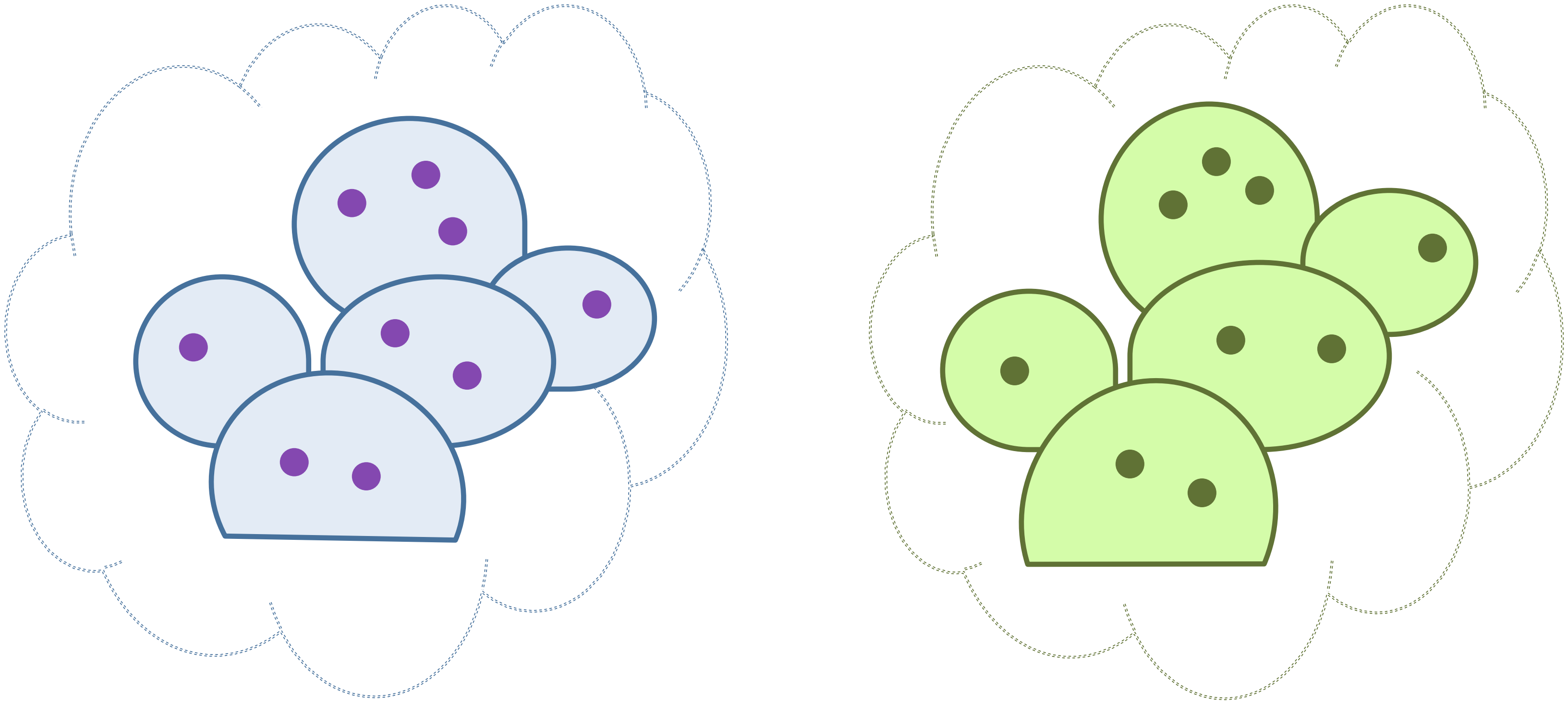
$$\mathcal{E} \sim \mathcal{F} \quad \text{iff} \quad (\emptyset, \emptyset) \in R$$

(Interleaving) Bisimulation

- Interleaving equivalence



Step Bisimulation



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

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

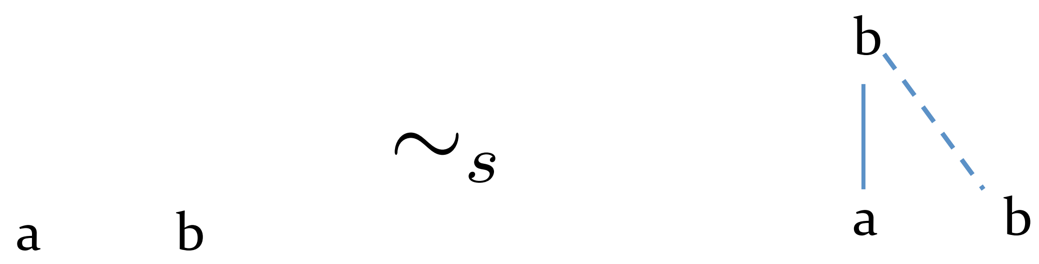
and X, X' are isomorphic steps (i.e., sets of concurrent events)

Step Bisimulation

- It observes concurrency

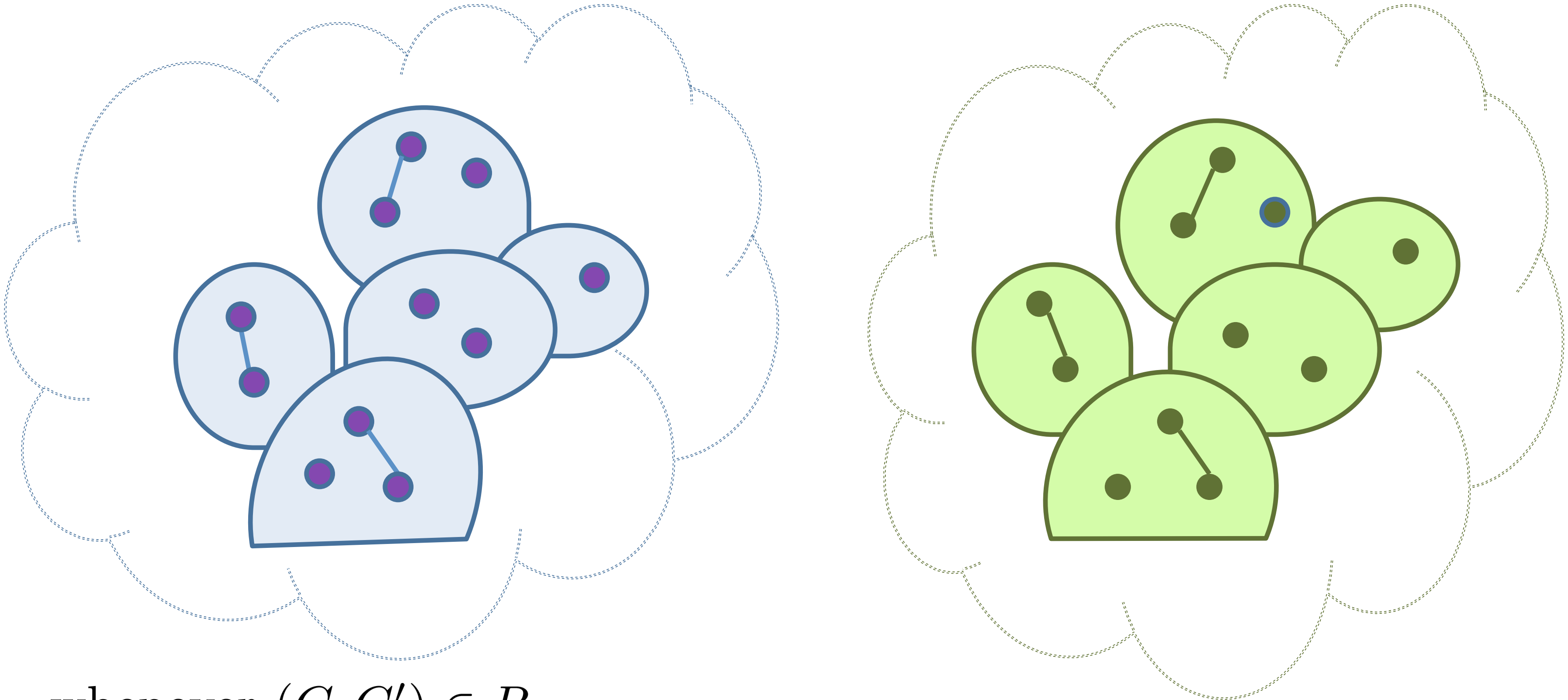
$$a \mid b \quad \not\sim_s \quad a.b + b.a$$

- but it cannot observe causality:



there is an occurrence of b
causally dependent from a

Pomset Bisimulation



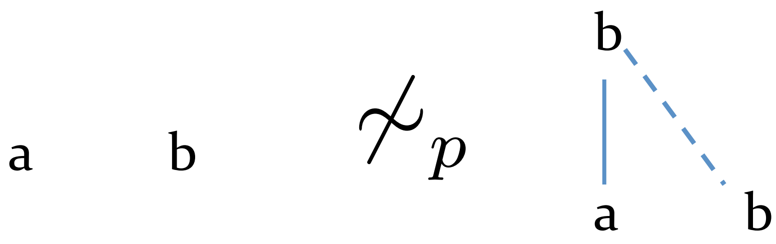
whenever $(C, C') \in R$

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

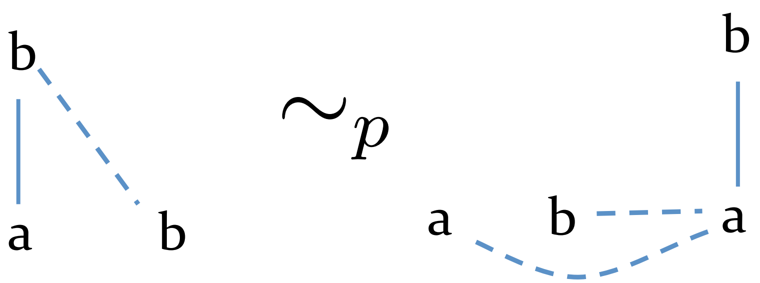
and X, X' are isomorphic pomsets (i.e., p.o. consistent events)

Pomset Bisimulation

- It captures causality



- but it cannot observe the causality / branching interplay:



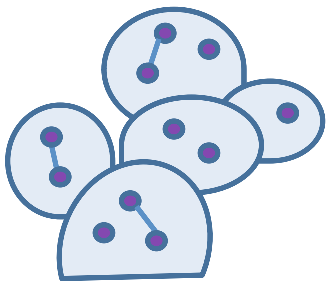
The same pomsets but only in the lhs

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

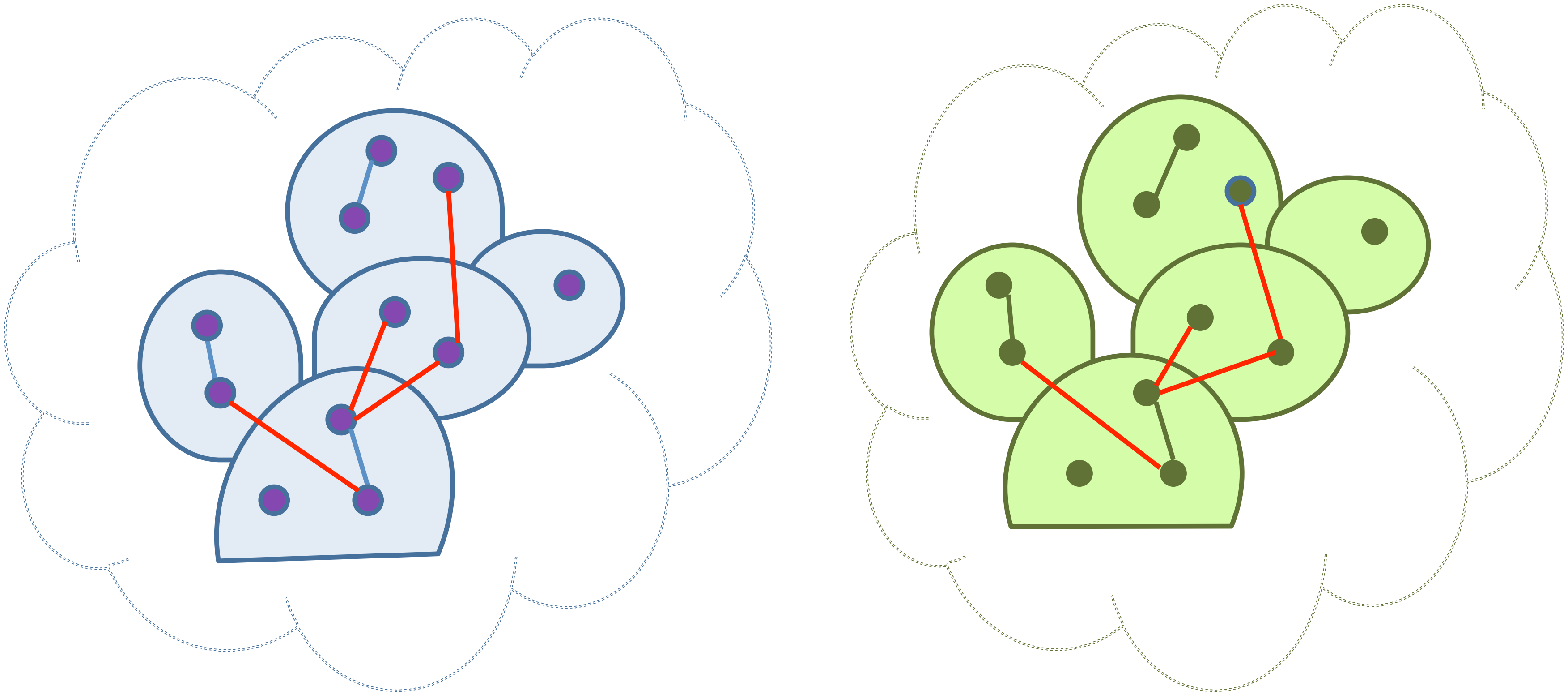
Pomset Bisimulation

- Analogously to bisimulation:
  - interleaving of pomsets (rather than actions)
  - it does not observe ***the dependencies between different pomset steps***

- keep the history of already matched transitions
  - Let the two matching configurations (entire history) in the game to be pomset-isomorphic
  - ***let the history grow pomset-isomorphically***



History-preserving Bisimulation



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

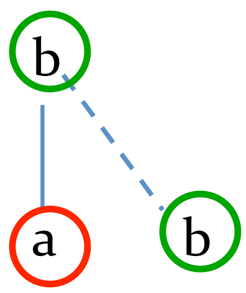
if $C \xrightarrow{e} D$ then $C' \xrightarrow{e'} D'$ with $(D, f[e \to e'], D') \in R$

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

History-preserving Bisimulation

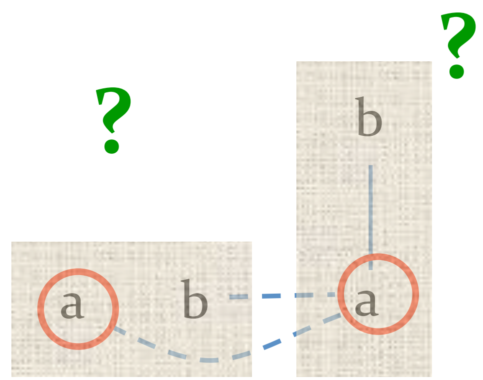
- It captures the causality / branching interplay

"causal bisimilarity"



$\sim_p$

$\not\sim_{hp}$



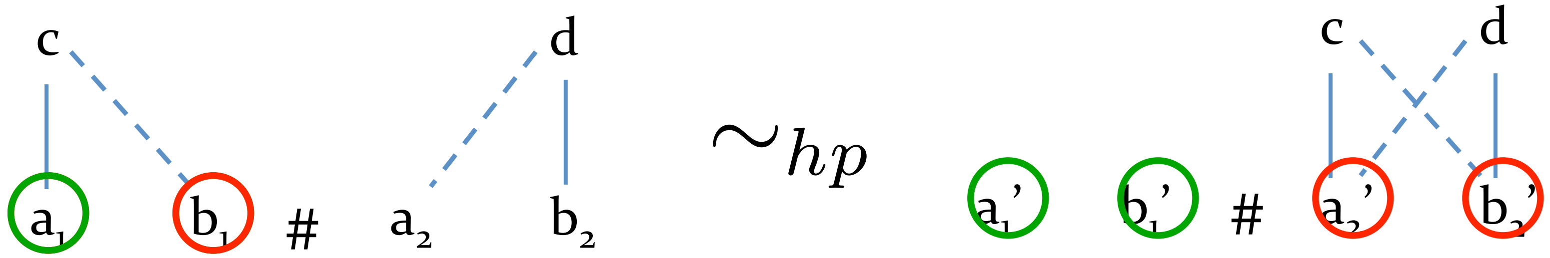
- It does not capture the interplay between causality – concurrency - branching

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 ***match depends on the order in which they are linearized*** (a_1, b_1 are concurrent)
- a_1, b_1 are inc
  "behavioral

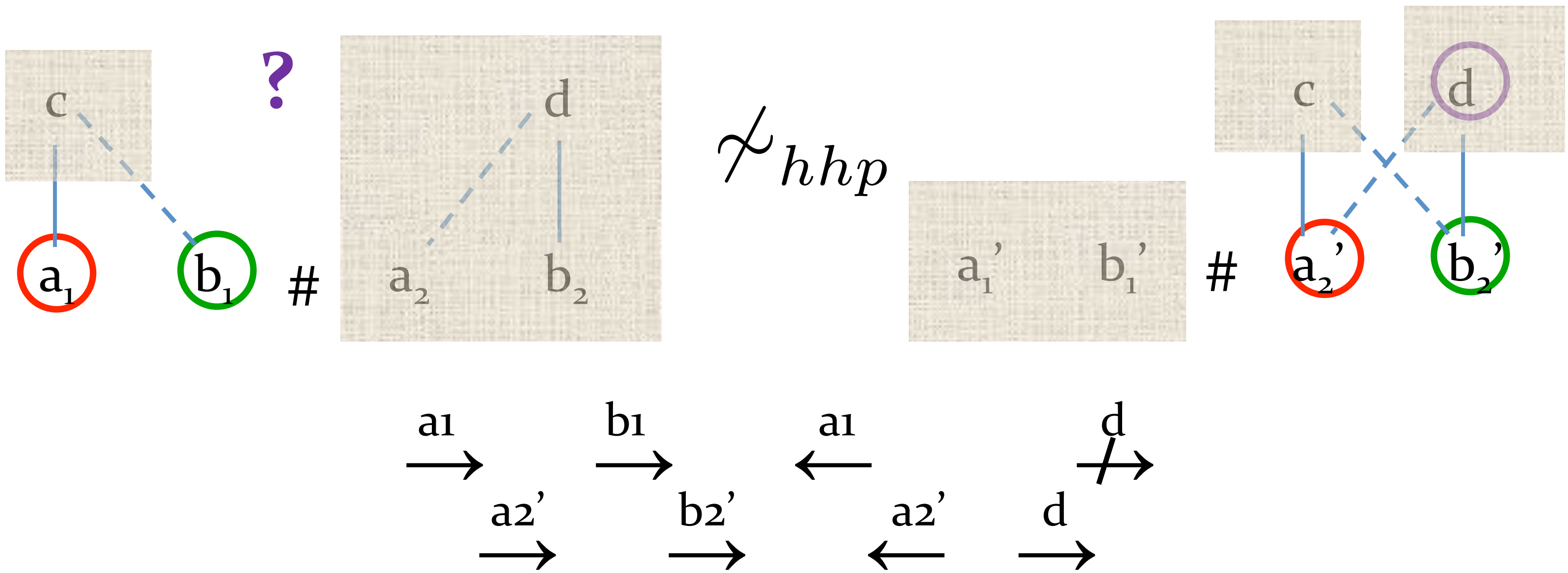
How can we formalize this difference?

Hereditary History-preserving Bisimulation

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

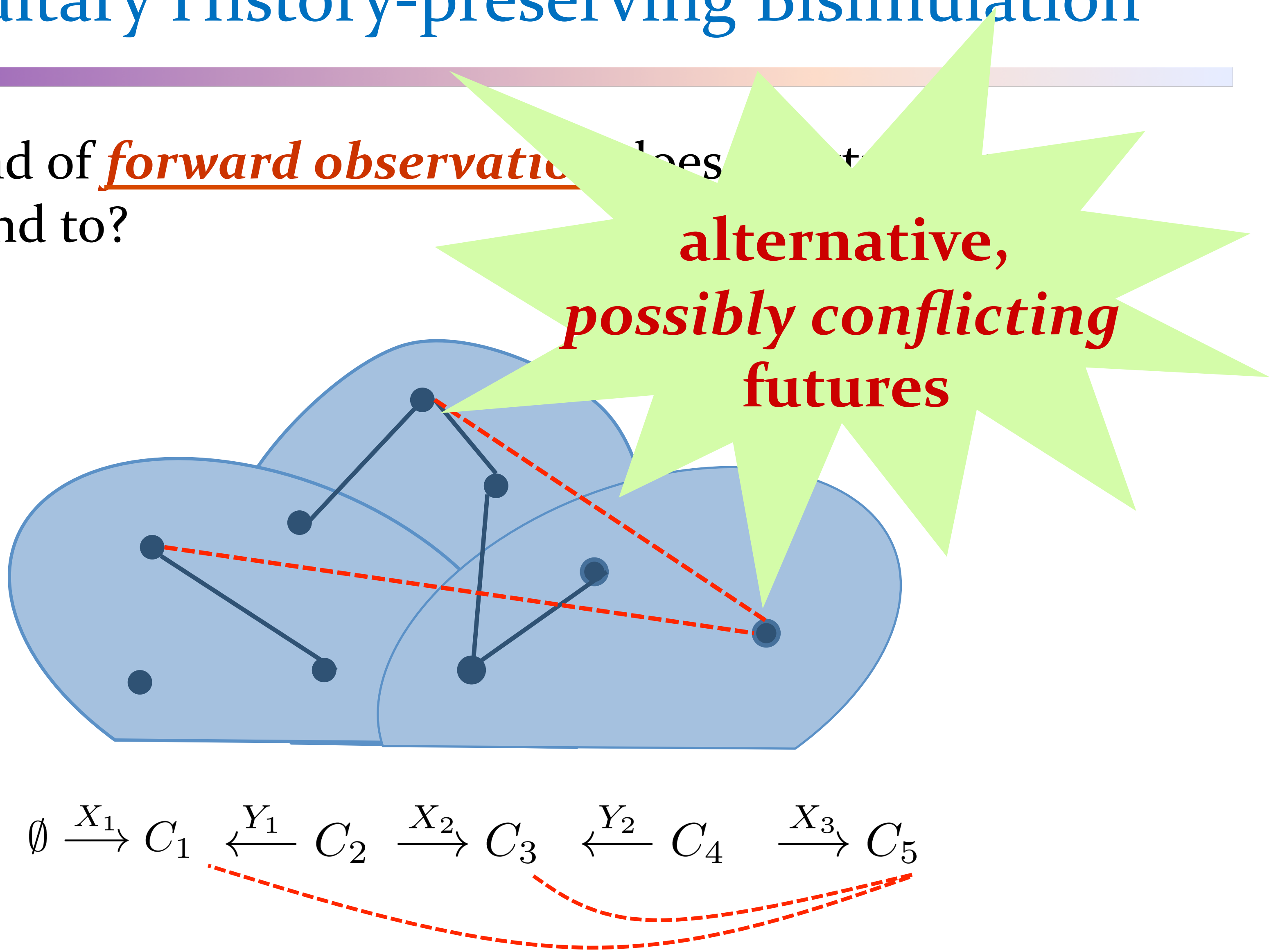
- if $C \xrightarrow{e} D$ then $C' \xrightarrow{e'} D'$ with $(D, f[e \to e'], D') \in R$
- if $D \xrightarrow{e} C$ then $D' \xrightarrow{e'} C'$ with $(D, f|_D, D') \in R$

Backward moves!!



Hereditary History-preserving Bisimulation

What kind of ***forward observatio*** ... does ... correspond to?



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

A logic for true concurrency

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

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 prime event structures:

$$C \models_{\eta} \varphi$$

configuration
current state of the computation

η : *Var* → E
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

z bound 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.

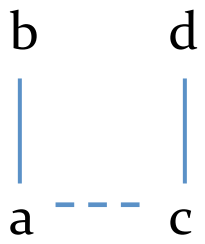
$$\eta(\mathbf{x}) < e, \ \eta(\mathbf{y}) || \, e, \ \lambda(e) = \mathsf{a} \text{ and } C \models_{\eta[z \to e]} \varphi$$

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

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

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

A logic for true concurrency



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

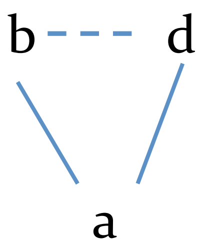
there is a future evolution that enables **b**

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

there are two (incompatible) futures

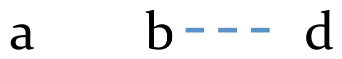
$$\emptyset \not\models_{\emptyset} (\mathsf{a}\, z)\langle z\rangle\, ((\mathsf{b}\, x) \wedge (\mathsf{d}\, y))$$

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 nation

- Immediate execution

sta

im

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

$$(\langle\!|\,\mathsf{a}\,|\!\rangle \otimes \langle\!|\,\mathsf{b}\,|\!\rangle \otimes \langle\!|\,\mathsf{c}\,|\!\rangle)\ \varphi$$

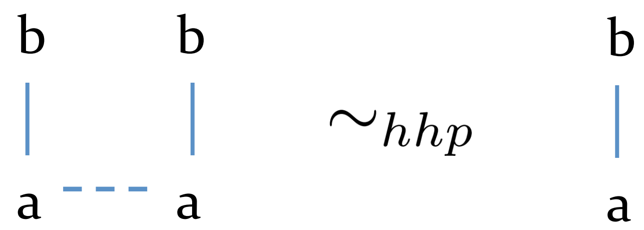
$$\underline{(\langle\!|\mathsf{a}\,x|\!\rangle \otimes \langle\!|\mathsf{a}\,y|\!\rangle)}\,(\underline{\langle\!|x < \mathsf{b}\,|\!\rangle \otimes \langle\!|\overline{y} < \mathsf{b}\,|\!\rangle})\,\varphi$$

- Step

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 $\mathcal{L}$:**

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

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

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
↓		↓
$\mathcal{L}_{hp}$		hp bisim
↓		↓
$\mathcal{L}_{p}$	⟺	Pomset bisim
↓		↓
$\mathcal{L}_{s}$		Step bisim
↓		↓
Hennessy-Milner Logic		Bisimulation equiv.
⇓		⇓

Logical Spectrum: HM Logic

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

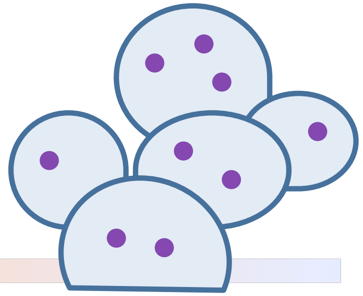
$$\varphi \;::=\; \langle\!|\mathsf{a}\; x|\!\rangle\, \varphi \mid \varphi \wedge \varphi \mid \neg\varphi \mid \top$$

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

<u>**Theorem:**</u> $\mathcal{E}_1 \equiv_{\mathcal{L}_{HM}} \mathcal{E}_2 \quad \text{iff} \quad \mathcal{E}_1 \sim \mathcal{E}_2$

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

Logical Spectrum: Step Logic



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

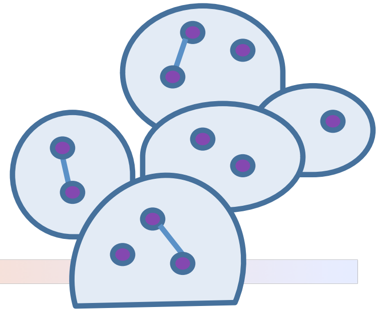
$$\varphi \;::=\; (\langle\!|\, \mathsf{a}_1\, x_1 \,|\!\rangle \otimes \cdots \otimes \langle\!|\, \mathsf{a}_n\, x_n \,|\!\rangle)\,\varphi \mid \varphi \wedge \varphi \mid \neg\varphi \mid \top$$

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

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

The logical equivalence induced by $\mathcal{L}_{\mathrm{s}}$ is step bisimulation

Logical Spectrum: Pomset Logic



The fragment $\mathcal{L}_p$:

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

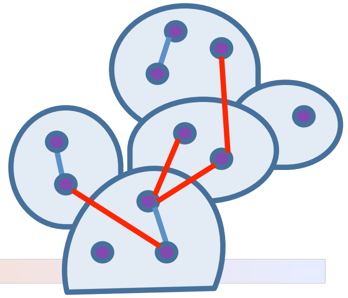
where ¬ , ∧ are used only **on closed formulae**

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

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

The logical equivalence induced by $\mathcal{L}_p$ is pomset bisimulation

Logical Spectrum: History Preserving Logic



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

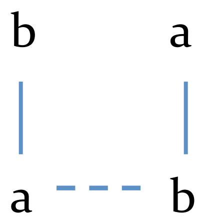
$$\varphi \;::=\; \langle\!|\mathbf{x}, \overline{\mathbf{y}} < \mathsf{a}\, z|\!\rangle\, \varphi \mid \varphi \wedge \varphi \mid \neg\varphi \mid \top$$

- Besides pomset execution, it also predicates about its dependencies with previously executed events
- **quantify + execute $\rightarrow$ no quantification over conflicting events**

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

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

Logical Spectrum: Separation Examples



$\sim$

$\not\sim_s$

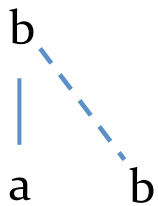
a b

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

a b

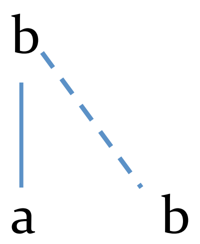
$\sim_s$

$\not\sim_p$



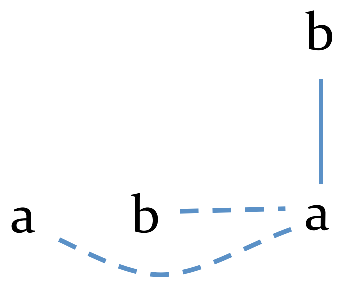
$$\mathcal{E}_1 \not\models, \mathcal{E}_2 \models \quad \langle\!|\mathsf{a}\; x|\!\rangle \langle\!| x < \mathsf{b}\; y|\!\rangle \qquad \in \mathcal{L}_p$$

Logical Spectrum: Separation Examples



$\sim_p$

$\not\sim_{hp}$

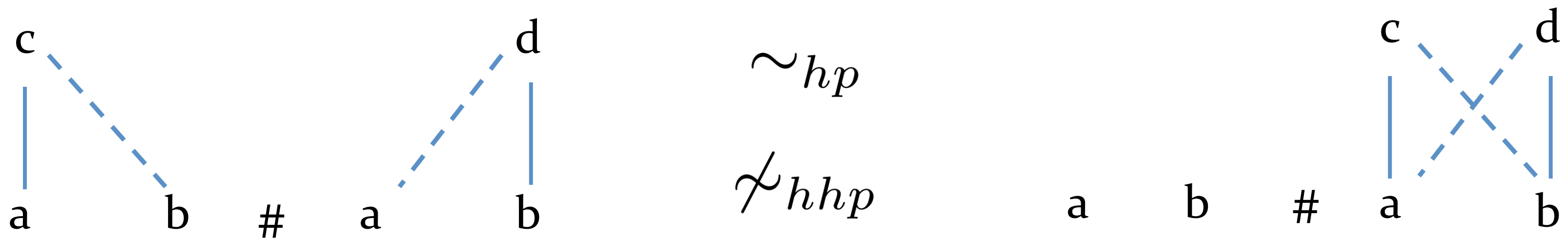


The same pomsets but only in the lhs

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

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

Logical Spectrum: Separation Examples



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

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

observe without executing:

conflicting futures

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

Future work

A unitary logical framework for true concurrent equivalences

- ***Study the logical true concurrent spectrum:***
  - linear time concurrent equivalences (trace/simulation hp, …)
  - observe without executing, but only predicate on consistent futures lies in between hp- and hhp-bis.

- ***Decidability border***
  - hp is decidable and hhp is undecidable for finite state systems. Characterise decidable equiv.

- ***Speicification logic***
  - 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

Future work

- ***Relation with other logic for concurrency:***
  - Past tense modality
- ***Proof theory***
- ***Model checking***
  - Automata- and game-theoretic approaches