Branching pomsets and event structures (oral communication)

Luc Edixhoven[1,2] Sung-Shik Jongmans[1,2]

José Proença[3] Ilaria Castellani[4]

[1]Open University of the Netherlands [2]CWI

[3]CISTER, ISEP, Polytechnic Institute of Porto [4]INRIA, Université Côte d'Azur

ICE 2023

Some context

Branching pomsets for choreographies

Luc Edixhoven[1,2] Guillermina Cledou[3,4]

Sung-Shik Jongmans[1,2] José Proença[5]

[1] Open University of the Netherlands [2] CWI

[3] HASLab, INESC TEC [4] University of Minho

[5] CISTER, ISEP, Polytechnic Institute of Porto

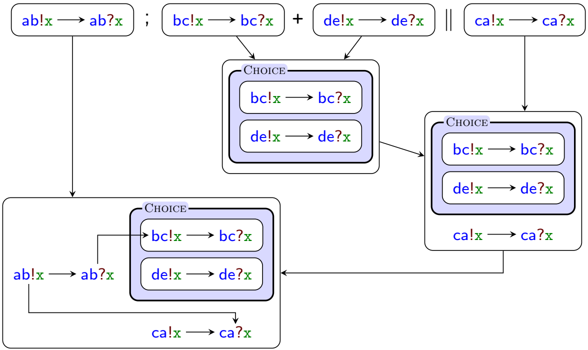
ICE 2022

1 / 23

Some context

Branching pomsets for choreographies

$$[\![a \rightarrow b{:}x \,;\, ((b \rightarrow c{:}x + d \rightarrow e{:}x) \parallel c \rightarrow a{:}x)]\!]$$



Some context

Conclusions and future work

Summary

- Branching pomsets
- Compact for both concurrency and choice
- Can express the same behaviour as choreographies

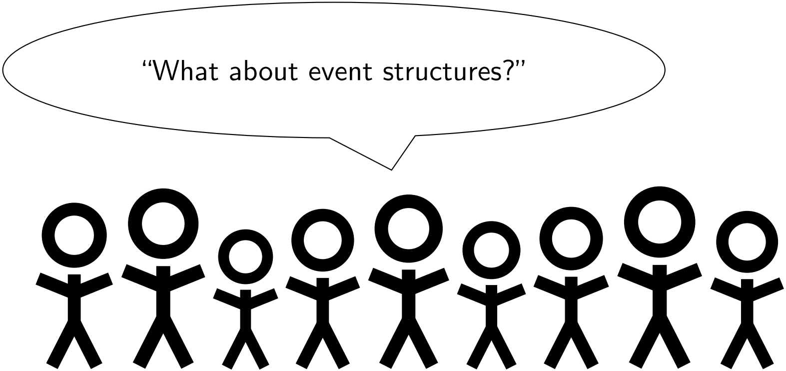
Future work

- Framework improvements: *n*-ary choices, partial order, loops
- Static analysis: realisability

`https://lmf.di.uminho.pt/b-pomset/`

23 / 23

Some context



Some context

Branching pomsets and event structures
(oral communication)

Luc Edixhoven[1,2] Sung-Shik Jongmans[1,2]

José Proença[3] Ilaria Castellani[4]

[1]Open University of the Netherlands [2]CWI

[3]CISTER, ISEP, Polytechnic Institute of Porto [4]INRIA, Université Côte d'Azur

ICE 2023

Outline

- **Branching pomsets**: a generic model for concurrency
- **Event structures**: a brief overview of the landscape
- **Comparison**: relative expressiveness

Branching pomsets and event structures

Basis: partially ordered multisets / pomsets (Pratt 1986)

$$
\begin{tikzcd}
a \arrow[r] \arrow[d] & c \arrow[r] & e \arrow[r] & g \arrow[d] \\
b & d \arrow[r] \arrow[ur] & f \arrow[r] & h
\end{tikzcd}
$$

- a <u>set of events</u>

  above: $\{a, b, c, d, e, f, g, h\}$

- a <u>partial order</u> on the events

  above: the reflexive and transitive closure of the arrows

- a <u>labelling function</u> from events to some set of labels

  above: omitted / identity (irrelevant for this talk)

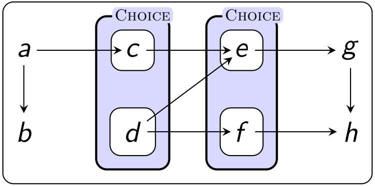
Branching pomsets and event structures

Extension: choices

- expressing choices with pomsets requires a set of pomsets
- with many choices, this set may become exponentially large
- solution: add a representation of choices

Branching pomsets

Choice model: branching structure



- add branching structure; a tree whose leaves are the events

  above: $\{a, b, g, h, \mathcal{C}_1, \mathcal{C}_2\}$,
  where $\mathcal{C}_1 = \{\{c\}, \{d\}\}$ and $\mathcal{C}_2 = \{\{e\}, \{f\}\}$

- replace the partial order with a precedence relation, whose reflexive and transitive closure is a partial order

  above: the arrows

Branching pomsets

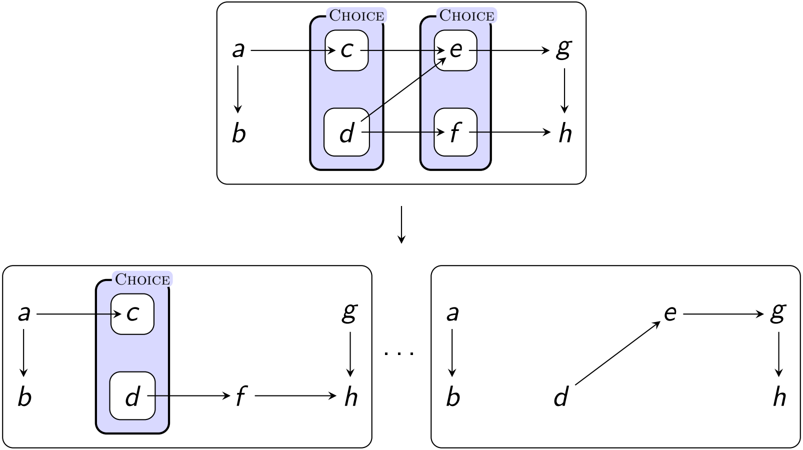
For comparison: the corresponding set of pomsets

$$\begin{tikzcd}
a \arrow[r] \arrow[d] & c \arrow[r] & e \arrow[r] & g \arrow[d] \\
b & & & h
\end{tikzcd}
\qquad
\begin{tikzcd}
a \arrow[d] & & e \arrow[r] & g \arrow[d] \\
b & d \arrow[ur] & & h
\end{tikzcd}$$

$$\begin{tikzcd}
a \arrow[r] \arrow[d] & c & & g \arrow[d] \\
b & & f \arrow[r] & h
\end{tikzcd}
\qquad
\begin{tikzcd}
a \arrow[d] & & & g \arrow[d] \\
b & d \arrow[r] & f \arrow[r] & h
\end{tikzcd}$$

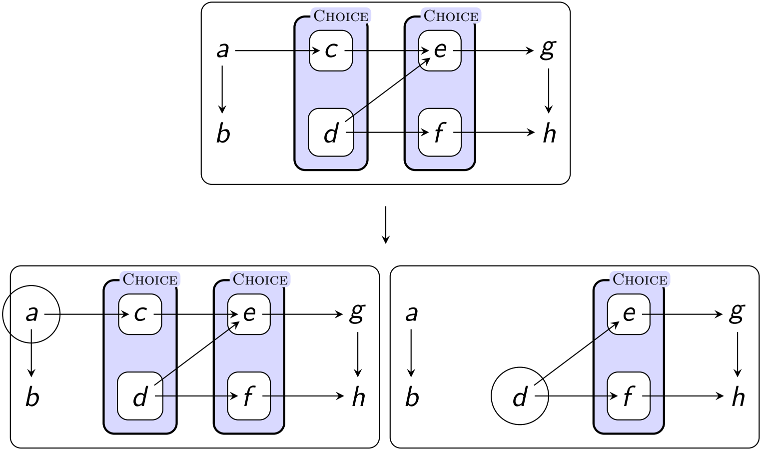
Branching pomsets

Semantics: refining $\Rightarrow$ resolving any number of choices



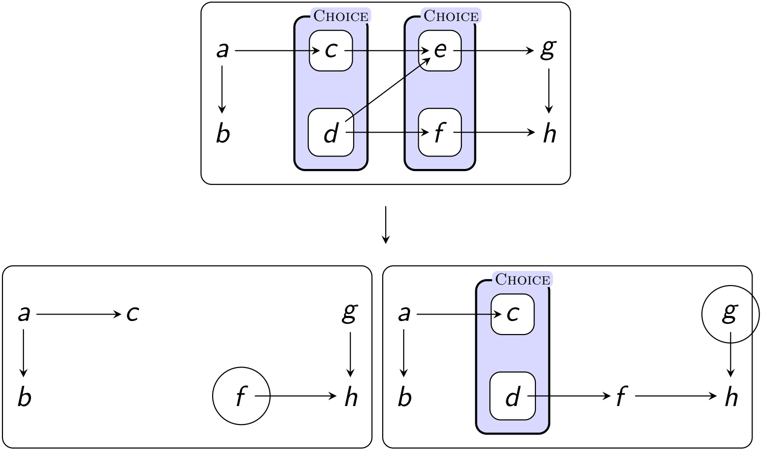
Branching pomsets

Semantics: enabling (followed by firing) ⇒ refining s.t. the chosen event is minimal and top-level, resolving no more than necessary



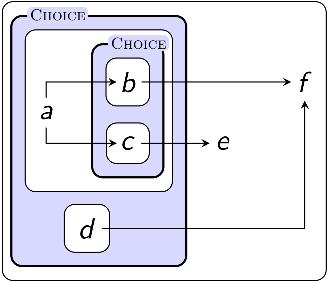
Branching pomsets

Semantics: enabling (followed by firing) $\Rightarrow$ refining s.t. the chosen event is minimal and top-level, resolving no more than necessary



Branching pomsets

Also: nested choices



Event structures

Choice model: conflict relation

$$\begin{tikzcd} a \arrow[r] \arrow[d] & c \arrow[r] \arrow[d, dotted, no head, "\#" description] & e \arrow[r] \arrow[d, dotted, no head, "\#" description] & g \arrow[d] \\ b & d \arrow[ur] \arrow[r] & f \arrow[r] & h \end{tikzcd}$$

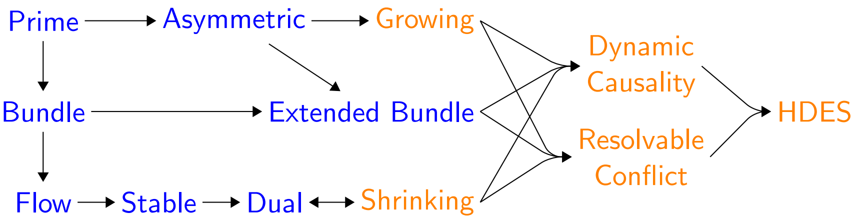
- add <u>conflict relation</u>; two conflicting events may not occur together in the same execution

  above: $\{(c, d), (e, f)\}$

- most classes of event structures define variations on causality and/or conflicts

Event structures

Landscape (partial): static and dynamic classes of event structures

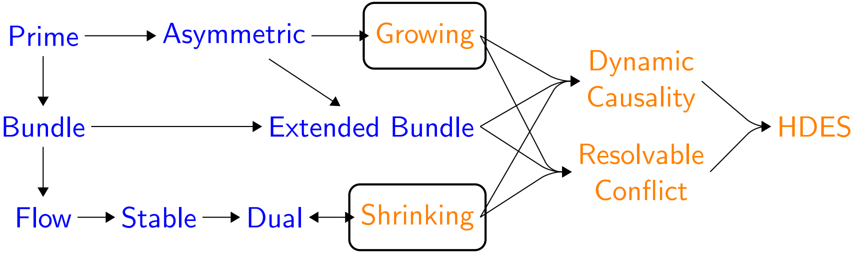


Arrows represent (strict) inclusion in terms of expressiveness

Figure: Arbach, Karcher, Peters and Nestmann, Dynamic causality in event structures (2018)

Event structures

Landscape (partial): static and dynamic classes of event structures

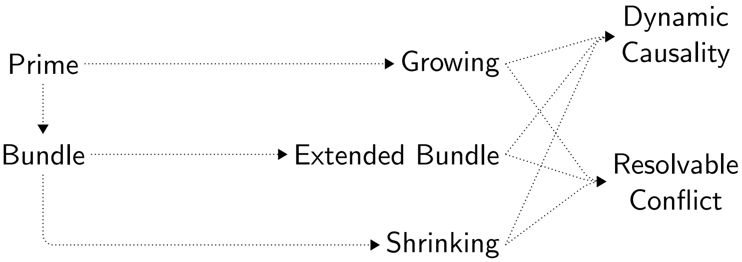


Arrows represent (strict) inclusion in terms of expressiveness

Figure: Arbach, Karcher, Peters and Nestmann, Dynamic causality in event structures (2018)

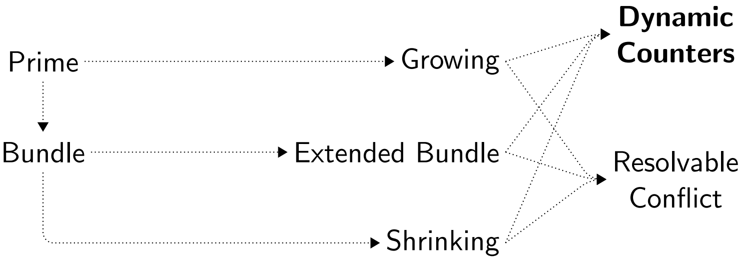
Most relevant for this talk: growing and shrinking causality $\Rightarrow$ dynamically adding and removing causalities

Comparison



Prime
Growing
Bundle
Extended Bundle
Shrinking
Dynamic Causality
Resolvable Conflict

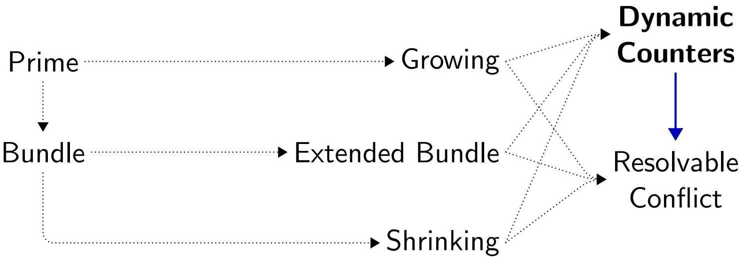
Comparison



Dynamic causality with counters: replaced dynamic causality event structures with a new variant with nice property; the order of events is irrelevant for the resulting causal state

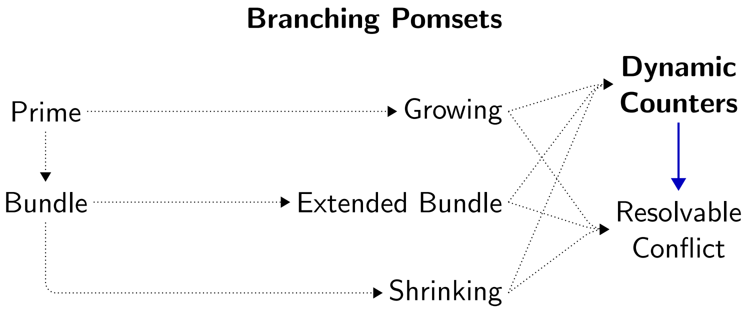
As a result: uniformly defined semantics for all shown classes

Comparison



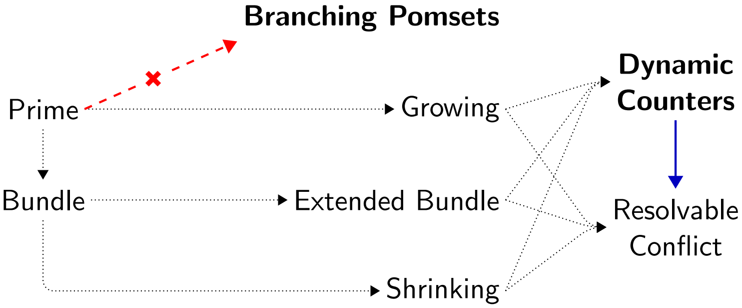
Generic proof: inclusion in event structures for resolvable conflict of any class of event structures where the causal state is order-independent, including dynamic counters

Comparison

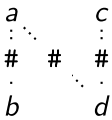


Next up: branching pomsets

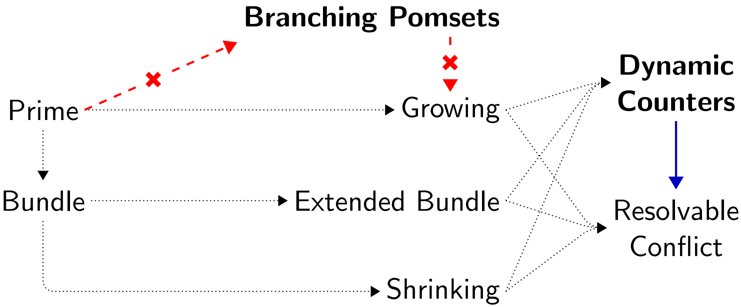
Comparison



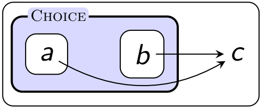
Non-inclusion: not all prime event structures expressible as branching pomsets — would need overlapping boxes



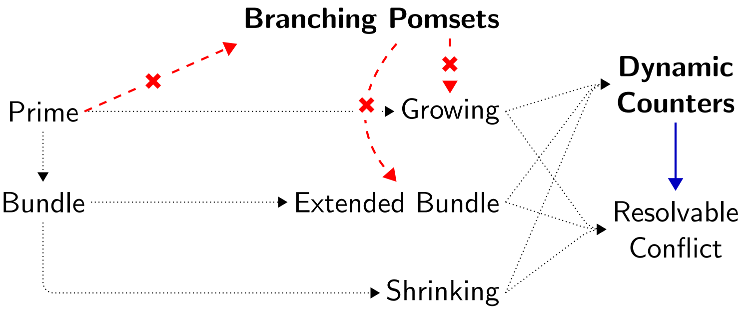
Comparison



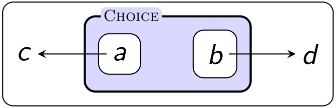
Non-inclusion: not all branching pomsets expressible as growing causality event structures — would need disjunctive causality



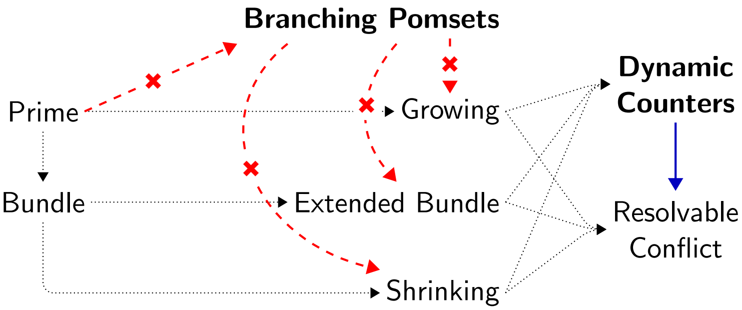
Comparison



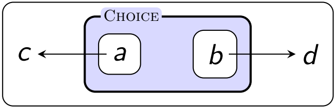
Non-inclusion: not all branching pomsets expressible as extended bundle event structures — c can be disabled and then re-enabled



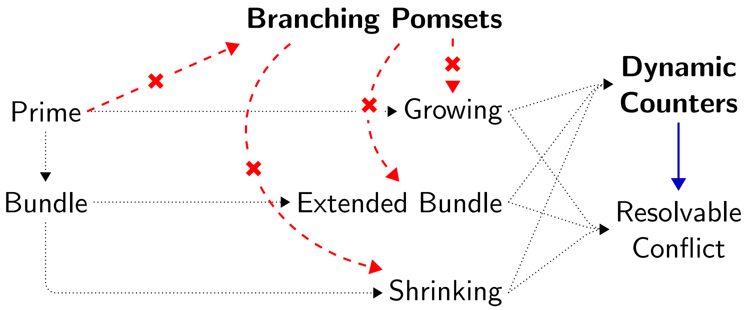
Comparison



Non-inclusion: not all branching pomsets expressible as shrinking causality event structures — c can be disabled and then re-enabled

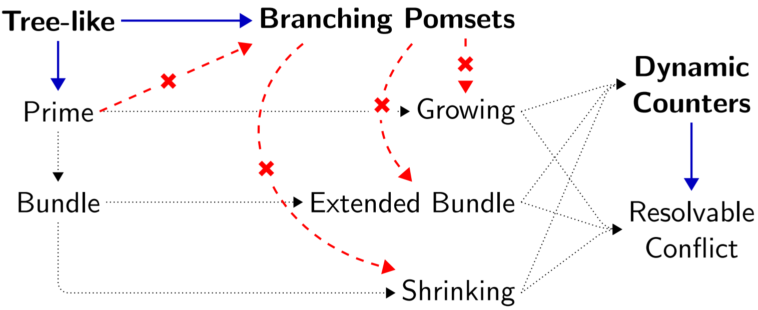


Comparison



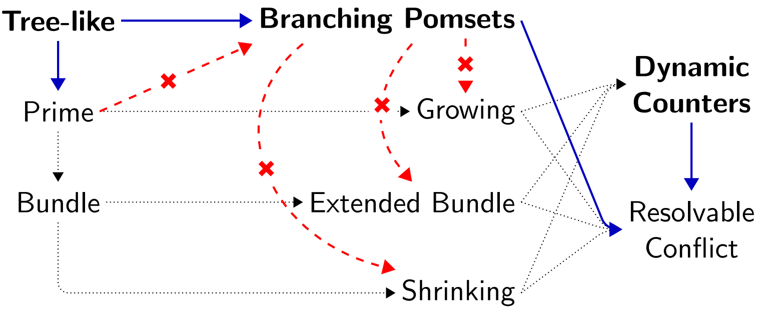
Consequently: branching pomsets incomparable with prime, growing causality, extended bundle and shrinking causality event structures

Comparison



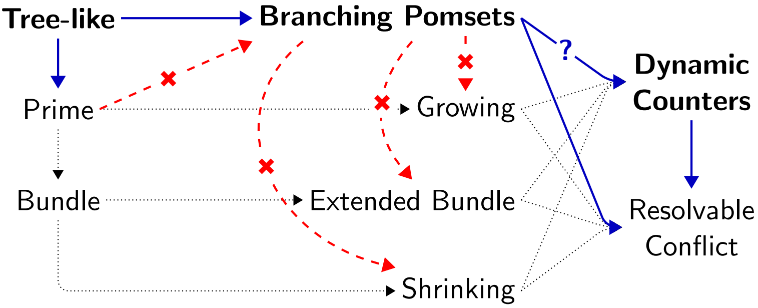
Inclusion: subset of branching pomsets, dubbed *tree-like*, can be expressed as prime event structures

Comparison



Inclusion: same generic proof as for event structures also holds for branching pomsets; they can all be expressed as event structures for resolvable conflict

Comparison



Inclusion conjecture: dynamic causality event structures (with counters) may be powerful enough to express all branching pomsets; no proof yet

Conclusions and future work

Summary

- branching pomsets as a generic model for concurrency
- comparison with various classes of event structures
- interesting behaviour: incomparable with most, included in some more expressive classes of dynamic event structures

Future work

- proving or disproving the dynamic counters conjecture
- study the expressiveness of branching pomsets with overlapping boxes
- expand static analysis of branching pomsets