How Reversibility Can Solve Traditional Questions: The Example of Hereditary History-Preserving Bisimulation

CONCUR 2020
The 31st International Conference on Concurrency Theory

Clément Aubert[1] Ioana Cristescu[2]

[1] Augusta University, School of Computer & Cyber Sciences, GA, USA

[2] Tarides, Paris

~~Vienna, Austria~~ — ONLINE — September 1st, 2020

In a nutshell

This work offers the characterization of
a relation
coming from
a denotationel model
in
a concurrent (reversible) calculus.

In a nutshell

This work offers the characterization of
Hereditary History-Preserving Bisimulation (HHPB)
coming from
a denotationel model
in
a concurrent (reversible) calculus.

In a nutshell

This work offers the characterization of
Hereditary History-Preserving Bisimulation (HHPB)
coming from
Labelled Configuration Structures*
in
a concurrent (reversible) calculus.

* a.k.a. Stable configuration structures, completed stable families.

Clément Aubert, Ioana Cristescu

In a nutshell

This work offers the characterization of
Hereditary History-Preserving Bisimulation (HHPB)
coming from
Labelled Configuration Structures*
in
Reversible Calculus of Communicating Systems (RCCS).

* a.k.a. Stable configuration structures, completed stable families.

In a nutshell

This work offers the characterization of
Hereditary History-Preserving Bisimulation (HHPB)
coming from
Labelled Configuration Structures*
in
Reversible Calculus of Communicating Systems (RCCS).

* a.k.a. Stable configuration structures, completed stable families.

And we have learned a thing on two on reversibility doing so.

Concurrent calculus

Study of *behaviour*.

Concurrent calculus

Study of *behaviour*.

Good calculus	Interesting way(s) of equating similar behaviors

Concurrent calculus

Study of *behaviour*.

Good calculus	Interesting way(s) of equating similar behaviors
CCS	Bissimulation
	Weak Bissimulation

Concurrent calculus

Study of *behaviour*.

Good calculus	Interesting way(s) of equating similar behaviors
CCS	Bissimulation
	Weak Bissimulation
RCCS / CCSK	?

Concurrent calculus

Study of *behaviour*.

Good calculus	Interesting way(s) of equating similar behaviors
CCS	Bissimulation
	Weak Bissimulation
RCCS / CCSK	?

Good models	Interesting way(s) of equating similar behaviors
Conf. Structures	Hereditary History-Preserving Bisimulation (HHPB)

Concurrent calculus

Study of *behaviour*.

Good calculus	Interesting way(s) of equating similar behaviors
CCS	Bissimulation
	Weak Bissimulation
RCCS	Back-and-forth Bisimulation (B&F)

Good models	Interesting way(s) of equating similar behaviors
Conf. Structures	Hereditary History-Preserving Bisimulation (HHPB)

How Reversibility Can Solve Traditional Questions: The Example of HHPB

Concurrent calculus

Study of *behaviour*.

Good calculus	Interesting way(s) of equating similar behaviors
CCS	Bissimulation
	Weak Bissimulation
RCCS	Back-and-forth Bisimulation (B&F)

Good models	Interesting way(s) of equating similar behaviors
Conf. Structures	Hereditary History-Preserving Bisimulation (HHPB)

Our result

$HHPB = B\&F$

A labeled configuration structure $\mathcal{C} = (E, C, L, \ell)$ is

- events $E = \{e_1, e_2, \dots\}$
- configurations $C = \{x, y, \dots\} \subseteq \wp(E)$
- labels $L = \{a, b, \tau, \dots\}$
- a labeling function $\ell : E \to L$

respecting *Finiteness*, *Coincidence Freenes*, *Finite Completeness* and *Stability*.

A labeled configuration structure $\mathcal{C} = (E, C, L, \ell)$ is

- events $E = \{e_1, e_2, \dots\}$
- configurations $C = \{x, y, \dots\} \subseteq \wp(E)$
- labels $L = \{a, b, \tau, \dots\}$
- a labeling function $\ell : E \to L$

respecting *Finiteness*, *Coincidence Freenes*, *Finite Completeness* and *Stability*.

$$\begin{tikzcd}
\{e_1, e_2\} & & \{e_1, e_3\} \\
& \{e_1\} \arrow[ul] \arrow[ur] & \\
& \emptyset \arrow[u] &
\end{tikzcd}
\qquad
\begin{aligned}
\ell = e_1 &\mapsto a, \\
e_2 &\mapsto b, \\
e_3 &\mapsto b.
\end{aligned}$$

A labeled configuration structure $\mathcal{C} = (E, C, L, \ell)$ is

- events $E = \{e_1, e_2, \dots\}$
- configurations $C = \{x, y, \dots\} \subseteq \wp(E)$
- labels $L = \{a, b, \tau, \dots\}$
- a labeling function $\ell : E \to L$

respecting *Finiteness*, *Coincidence Freenes*, *Finite Completeness* and *Stability*.

$$\begin{tikzcd}[column sep=small]
\{a, b_1\} & & \{a, b_2\} \\
& \{a\} \arrow[ul] \arrow[ur] & \\
& \emptyset \arrow[u] &
\end{tikzcd}$$

A labeled configuration structure $\mathcal{C} = (E, C, L, \ell)$ is

- events $E = \{e_1, e_2, \ldots\}$
- labels $L = \{a, b, \tau, \ldots\}$
- configurations $C = \{x, y, \ldots\} \subseteq \wp(E)$
- a labeling function $\ell : E \to L$

respecting *Finiteness*, *Coincidence Freenes*, *Finite Completeness* and *Stability*.

$$\begin{tikzcd} \{a, b_1\} & & \{a, b_2\} \\ & \{a\} \arrow[ul] \arrow[ur] & \\ & \emptyset \arrow[u] & \end{tikzcd}$$

$$\emptyset \xrightarrow{a} \{a\}$$

$$\{a\} \xrightarrow{b_1} \{a, b_1\}$$

$$\{a\} \xrightarrow{b_2} \{a, b_2\}$$

A labeled configuration structure $\mathcal{C} = (E, C, L, \ell)$ is

- events $E = \{e_1, e_2, \dots\}$
- configurations $C = \{x, y, \dots\} \subseteq \wp(E)$
- labels $L = \{a, b, \tau, \dots\}$
- a labeling function $\ell : E \to L$

respecting *Finiteness*, *Coincidence Freenes*, *Finite Completeness* and *Stability*.

$$
\begin{tikzcd}
\{a, b_1\} & & \{a, b_2\} \\
& \{a\} \arrow[ul] \arrow[ur] & \\
& \emptyset \arrow[u] &
\end{tikzcd}
$$

$$\emptyset \overset{a}{\leftsquigarrow} \{a\}$$

$$\{a\} \overset{b_1}{\leftsquigarrow} \{a, b_1\}$$

$$\{a\} \overset{b_2}{\leftsquigarrow} \{a, b_2\}$$

A labeled configuration structure $\mathcal{C} = (E, C, L, \ell)$ is

- events $E = \{e_1, e_2, \dots\}$
- labels $L = \{a, b, \tau, \dots\}$
- configurations $C = \{x, y, \dots\} \subseteq \wp(E)$
- a labeling function $\ell : E \to L$

respecting *Finiteness*, *Coincidence Freenes*, *Finite Completeness* and *Stability*.

$$\begin{tikzcd}[column sep=small]
\{a, b_1\} & & \{a, b_2\} \\
& \{a\} \arrow[ul] \arrow[ur] & \\
& \varnothing \arrow[u] &
\end{tikzcd}$$

$$[\![a.(b + b)]\!]$$

A labeled configuration structure $\mathcal{C} = (E, C, L, \ell)$ is

- events $E = \{e_1, e_2, \dots\}$
- configurations $C = \{x, y, \dots\} \subseteq \wp(E)$
- labels $L = \{a, b, \tau, \dots\}$
- a labeling function $\ell : E \to L$

respecting *Finiteness*, *Coincidence Freenes*, *Finite Completeness* and *Stability*.

$$
\begin{tikzcd}
\{a, b_1\} & & \{a, b_2\} \\
& \{a\} \arrow[ul] \arrow[ur] & \\
& \emptyset \arrow[u] &
\end{tikzcd}
\qquad
\begin{tikzcd}
\{a_1, b_1\} & & \{a_2, b_2\} \\
\{a_1\} \arrow[u] & & \{a_2\} \arrow[u] \\
& \emptyset \arrow[ul] \arrow[ur] &
\end{tikzcd}
$$

$$\llbracket a.(b + b) \rrbracket \qquad \llbracket (a.b) + (a.b) \rrbracket$$

Clément Aubert, Ioana Cristescu

A labeled configuration structure $\mathcal{C} = (E, C, L, \ell)$ is

- events $E = \{e_1, e_2, \dots\}$
- labels $L = \{a, b, \tau, \dots\}$
- configurations $C = \{x, y, \dots\} \subseteq \wp(E)$
- a labeling function $\ell : E \to L$

respecting *Finiteness*, *Coincidence Freenes*, *Finite Completeness* and *Stability*.

$$\begin{tikzcd}
\{a, b_1\} & & \{a, b_2\} \\
& \{a\} \arrow[ul] \arrow[ur] & \\
& \emptyset \arrow[u] &
\end{tikzcd}
\qquad
\begin{tikzcd}
\{a_1, b_1\} & & \{a_2, b_2\} \\
\{a_1\} \arrow[u] & & \{a_2\} \arrow[u] \\
& \emptyset \arrow[ul] \arrow[ur] &
\end{tikzcd}
\qquad
\begin{tikzcd}
& \{a_1, a_2\} & \\
\{a_1\} \arrow[ur] & & \{a_2\} \arrow[ul] \\
& \emptyset \arrow[ul] \arrow[ur] &
\end{tikzcd}$$

$\llbracket a.(b + b) \rrbracket$ $\quad$ $\llbracket (a.b) + (a.b) \rrbracket$ $\quad$ $\llbracket a \mid a \rrbracket$

A labeled configuration structure $\mathcal{C} = (E, C, L, \ell)$ is

- events $E = \{e_1, e_2, \dots\}$
- configurations $C = \{x, y, \dots\} \subseteq \wp(E)$
- labels $L = \{a, b, \tau, \dots\}$
- a labeling function $\ell : E \to L$

respecting *Finiteness*, *Coincidence Freenes*, *Finite Completeness* and *Stability*.

$$\begin{tikzcd}
\{a, b_1\} & & \{a, b_2\} \\
& \{a\} \arrow[ul] \arrow[ur] & \\
& \emptyset \arrow[u] &
\end{tikzcd}$$

$[\![a.(b + b)]\!]$

$a \leqslant b_1$

$a \leqslant b_2$

$$\begin{tikzcd}
\{a_1, b_1\} & & \{a_2, b_2\} \\
\{a_1\} \arrow[u] & & \{a_2\} \arrow[u] \\
& \emptyset \arrow[ul] \arrow[ur] &
\end{tikzcd}$$

$[\![(a.b) + (a.b)]\!]$

$a_1 \leqslant b_1$

$a_2 \leqslant b_2$

$$\begin{tikzcd}
& \{a_1, a_2\} & \\
\{a_1\} \arrow[ur] & & \{a_2\} \arrow[ul] \\
& \emptyset \arrow[ul] \arrow[ur] &
\end{tikzcd}$$

$[\![a \mid a]\!]$

$\mathcal{C}_1$ and $\mathcal{C}_2$ are HHPB if

$\exists \mathcal{R} \subseteq C_1 \times C_2 \times (E_1 \rightharpoonup E_2)$ such that

Clément Aubert, Ioana Cristescu

$\mathcal{C}_1$ and $\mathcal{C}_2$ are HHPB if

$\exists \mathcal{R} \subseteq C_1 \times C_2 \times (E_1 \rightharpoonup E_2)$ such that $(\emptyset, \emptyset, \emptyset) \in \mathcal{R}$,

$\mathcal{C}_1$ and $\mathcal{C}_2$ are HHPB if

$\exists \mathcal{R} \subseteq C_1 \times C_2 \times (E_1 \rightharpoonup E_2)$ such that $(\emptyset, \emptyset, \emptyset) \in \mathcal{R}$, and if $(x_1, x_2, f) \in \mathcal{R}$, then f is a label- and order- preserving bijection between x_1 and x_2

$\mathcal{C}_1$ and $\mathcal{C}_2$ are HHPB if

$\exists \mathcal{R} \subseteq C_1 \times C_2 \times (E_1 \rightharpoonup E_2)$ such that $(\emptyset, \emptyset, \emptyset) \in \mathcal{R}$, and if $(x_1, x_2, f) \in \mathcal{R}$, then f is a label- and order- preserving bijection between x_1 and x_2

$$\boxed{\begin{aligned} f \text{ is l\&o-p} \Rightarrow & \ell_1(e_1) = \ell_2(f(e_1)) \\ + & e_1 \leqslant e_2 \Rightarrow f(e_1) \leqslant f(e_2) \end{aligned} \quad \text{for all } e_1, e_2 \in x_1}$$

Clément Aubert, Ioana Cristescu

$\mathcal{C}_1$ and $\mathcal{C}_2$ are HHPB if

$\exists \mathcal{R} \subseteq C_1 \times C_2 \times (E_1 \rightharpoonup E_2)$ such that $(\emptyset, \emptyset, \emptyset) \in \mathcal{R}$, and if $(x_1, x_2, f) \in \mathcal{R}$, then f is a label- and order- preserving bijection between x_1 and x_2 such that:

$$\forall y_1, x_1 \xrightarrow{e_1} y_1 \Rightarrow \exists y_2, g, x_2 \xrightarrow{e_2} y_2, g\restriction_{x_1} = f, (y_1, y_2, g) \in \mathcal{R}$$

$\mathcal{C}_1$ and $\mathcal{C}_2$ are HHPB if

$\exists \mathcal{R} \subseteq C_1 \times C_2 \times (E_1 \rightharpoonup E_2)$ such that $(\varnothing, \varnothing, \varnothing) \in \mathcal{R}$, and if $(x_1, x_2, f) \in \mathcal{R}$, then f is a label- and order- preserving bijection between x_1 and x_2 such that:

$$\forall y_1, x_1 \xrightarrow{e_1} y_1 \Rightarrow \exists y_2, g, x_2 \xrightarrow{e_2} y_2, g\restriction_{x_1} = f, (y_1, y_2, g) \in \mathcal{R}$$

$$\forall y_2, x_2 \xrightarrow{e_2} y_2 \Rightarrow \exists y_1, g, x_1 \xrightarrow{e_1} y_1, g\restriction_{x_1} = f, (y_1, y_2, g) \in \mathcal{R}$$

$\mathcal{C}_1$ and $\mathcal{C}_2$ are HHPB if

$\exists \mathcal{R} \subseteq C_1 \times C_2 \times (E_1 \rightharpoonup E_2)$ such that $(\emptyset, \emptyset, \emptyset) \in \mathcal{R}$, and if $(x_1, x_2, f) \in \mathcal{R}$, then f is a label- and order- preserving bijection between x_1 and x_2 such that:

$$\forall y_1, x_1 \xrightarrow{e_1} y_1 \Rightarrow \exists y_2, g, x_2 \xrightarrow{e_2} y_2, g\restriction_{x_1} = f, (y_1, y_2, g) \in \mathcal{R}$$

$$\forall y_2, x_2 \xrightarrow{e_2} y_2 \Rightarrow \exists y_1, g, x_1 \xrightarrow{e_1} y_1, g\restriction_{x_1} = f, (y_1, y_2, g) \in \mathcal{R}$$

$$\forall y_1, x_1 \overset{e_1}{\rightsquigarrow} y_1 \Rightarrow \exists y_2, g, x_2 \overset{e_2}{\rightsquigarrow} y_2, g = f\restriction_{y_1}, (y_1, y_2, g) \in \mathcal{R}$$

$$\forall y_2, x_2 \overset{e_2}{\rightsquigarrow} y_2 \Rightarrow \exists y_1, g, x_1 \overset{e_1}{\rightsquigarrow} y_1, g = f\restriction_{y_1}, (y_1, y_2, g) \in \mathcal{R}$$

Example of structures in HHPB

$$
\begin{tikzcd}
\{a, b_1\} & & \{a, b_2\} \\
& \{a\} \arrow[ul] \arrow[ur] & \\
& \varnothing \arrow[u] &
\end{tikzcd}
\qquad
\begin{tikzcd}
\{a_1, b_1\} & & \{a_2, b_2\} \\
\{a_1\} \arrow[u] & & \{a_2\} \arrow[u] \\
& \varnothing \arrow[ul] \arrow[ur] &
\end{tikzcd}
$$

Example of structures in HHPB

$$\begin{tikzcd}[column sep=small]
\{a, b_1\} & & \{a, b_2\} & & \{a_1, b_1\} & & \{a_2, b_2\} \\
& \{a\} \arrow[ul] \arrow[ur] & & & \{a_1\} \arrow[u] & & \{a_2\} \arrow[u] \\
& \boxed{\varnothing} \arrow[u] & & & & \boxed{\varnothing} \arrow[ul] \arrow[ur] &
\end{tikzcd}$$

$$\mathcal{R} = \{(\varnothing, \varnothing, \varnothing), \qquad \}$$

Example of structures in HHPB

$$
\begin{tikzcd}[column sep=small]
\{a, b_1\} & & \{a, b_2\} \\
& \{a\} \arrow[ul] \arrow[ur] & \\
& \emptyset \arrow[u] &
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}[column sep=small]
\{a_1, b_1\} & & \{a_2, b_2\} \\
\{a_1\} \arrow[u] & & \{a_2\} \arrow[u] \\
& \emptyset \arrow[ul] \arrow[ur] &
\end{tikzcd}
$$

$$\mathcal{R} = \{(\emptyset, \emptyset, \emptyset),$$

$$\}$$

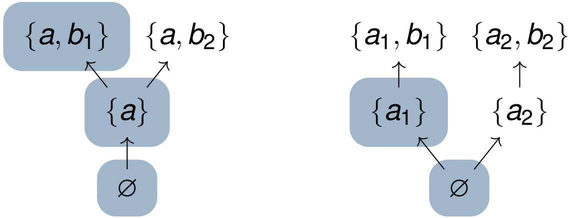
Example of structures in HHPB

$$
\begin{tikzcd}
\{a, b_1\} & & \{a, b_2\} \\
& \{a\} \arrow[ul] \arrow[ur] & \\
& \varnothing \arrow[u] &
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
\{a_1, b_1\} & & \{a_2, b_2\} \\
\{a_1\} \arrow[u] & & \{a_2\} \arrow[u] \\
& \varnothing \arrow[ul] \arrow[ur] &
\end{tikzcd}
$$

$$
\begin{aligned}
\mathcal{R} = \{ & (\varnothing, \varnothing, \varnothing), \\
& (\{a\}, \{a_1\}, \{a \mapsto a_1\}), \\
& \}
\end{aligned}
$$

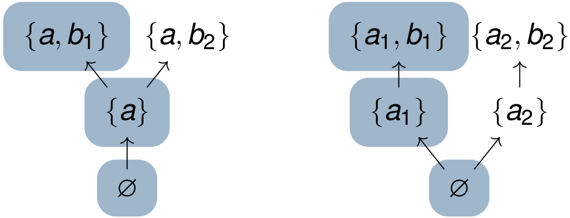
Clément Aubert, Ioana Cristescu

Example of structures in HHPB



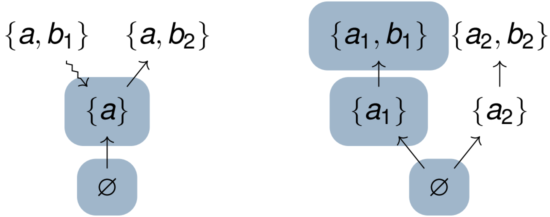
$$\begin{aligned}\mathcal{R} = \{&(\emptyset, \emptyset, \emptyset),\\ &(\{a\}, \{a_1\}, \{a \mapsto a_1\}),\\ &\}\end{aligned}$$

Example of structures in HHPB



$$
\begin{aligned}
\mathcal{R} = \{&(\emptyset, \emptyset, \emptyset),\\
&(\{a\}, \{a_1\}, \{a \mapsto a_1\}),\\
&(\{a, b_1\}, \{a_1, b_1\}, \{a \mapsto a_1, b_1 \mapsto b_1\}),\\
&\}
\end{aligned}
$$

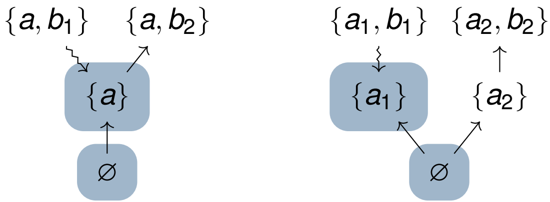
Example of structures in HHPB



$$
\begin{aligned}
\mathcal{R} = \{&(\emptyset, \emptyset, \emptyset),\\
&(\{a\}, \{a_1\}, \{a \mapsto a_1\}),\\
&(\{a, b_1\}, \{a_1, b_1\}, \{a \mapsto a_1, b_1 \mapsto b_1\}),\\
&\}
\end{aligned}
$$

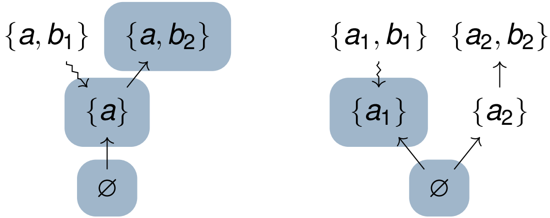
Clément Aubert, Ioana Cristescu

Example of structures in HHPB



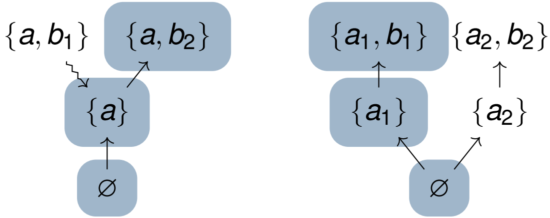
$$
\begin{aligned}
\mathcal{R} = \{ & (\varnothing, \varnothing, \varnothing), \\
& (\{a\}, \{a_1\}, \{a \mapsto a_1\}), \\
& (\{a, b_1\}, \{a_1, b_1\}, \{a \mapsto a_1, b_1 \mapsto b_1\}), \\
& \}
\end{aligned}
$$

Example of structures in HHPB



$$\begin{aligned}\mathcal{R} = \{&(\emptyset, \emptyset, \emptyset),\\ &(\{a\}, \{a_1\}, \{a \mapsto a_1\}),\\ &(\{a, b_1\}, \{a_1, b_1\}, \{a \mapsto a_1, b_1 \mapsto b_1\}),\\ &\}\end{aligned}$$

Example of structures in HHPB



$$
\begin{aligned}
\mathcal{R} = \{ & (\emptyset, \emptyset, \emptyset), \\
& (\{a\}, \{a_1\}, \{a \mapsto a_1\}), \\
& (\{a, b_1\}, \{a_1, b_1\}, \{a \mapsto a_1, b_1 \mapsto b_1\}), \\
& (\{a, b_2\}, \{a_1, b_1\}, \{a \mapsto a_1, b_2 \mapsto b_1\}), \cdots \}
\end{aligned}
$$

Why do we care about HHPB?

Acta Informatica 37, 229–327 (2001)

Acta Informatica

Refinement of actions and equivalence notions for concurrent systems

Rob van Glabbeek[1], **Ursula Goltz**[2]

conflict [Winskel] upgraded with a termination predicate. We argue that history preserving and hereditary history preserving equivalence both preserve causality, branching, and their interplay, and both abstract from choices between identical alternatives; however, the latter may be the finest reasonable equivalence with these properties – it thoroughly respects the internal structure of related systems – whereas the former may be the coarsest equivalence of this kind, still making nontrivial identifications.

Why do we care about HHPB?

INFORMATION AND COMPUTATION **127**, 164–185 (1996)
ARTICLE NO. 0057

Bisimulation from Open Maps

ANDRÉ JOYAL

Département de mathématiques et d'informatique, Université du Québec à Montréal, Montréal, Quebec Canada H3C 3P8

AND

MOGENS NIELSEN AND GLYNN WINSKEL

Computer Science Department, Aarhus University, 8000 Aarhus C, Denmark

An abstract definition of bisimulation is presented. It makes possible a uniform definition of bisimulation across a range of different models for parallel computation presented as categories. As examples, transition systems, synchronisation trees, transition systems with independence (an abstraction from Petri nets), and labelled event structures are considered. On transition systems the abstract definition readily specialises to Milner's strong bisimulation. On event structures it explains and leads to a strengthening of the history-preserving bisimulation of Rabinovitch and Traktenbrot and van Glabeek and

Why do we care about HHPB?

Available online at www.sciencedirect.com
ScienceDirect
ELSEVIER
Electronic Notes in Theoretical Computer Science 192 (2007) 93–108
Electronic Notes in Theoretical Computer Science
www.elsevier.com/locate/entcs

Reversibility and Models for Concurrency

Iain Phillips[1]
Department of Computing
Imperial College London
180 Queen's Gate, London, SW7 2AZ, United Kingdom

Irek Ulidowski[2]
Department of Computer Science
University of Leicester
University Road, Leicester, LE1 7RH, United Kingdom

tion law: $(a \mid (b+c)) + (a \mid b) + ((a+c) \mid b) = (a \mid (b+c)) + ((a+c) \mid b)$. We show that FR bisimulation coincides with *hereditary history-preserving (HHP) bisimulation*, which is regarded as the canonical true concurrency equivalence [1,5,7,3]. The result holds for reversible transition systems with no auto-concurrency and with no auto-causation, and since CCSK gives rise to such transition systems the result holds for CCSK.

Why do we care about HHPB?

ELSEVIER
Available online at www.sciencedirect.com
ScienceDirect
Electronic Notes in Theoretical Computer Science 192 (2007) 93–108
Electronic Notes in Theoretical Computer Science
www.elsevier.com/locate/entcs

Reversibility and Models for Concurrency

Iain Phillips[1]

Department of Computing
Imperial College London
180 Queen's Gate, London, SW7 2AZ, United Kingdom

Irek Ulidowski[2]

Department of Computer Science
University of Leicester
University Road, Leicester, LE1 7RH, United Kingdom

Why do we care about HHPB?

Available online at www.sciencedirect.com
ScienceDirect
Electronic Notes in Theoretical Computer Science
ELSEVIER
Electronic Notes in Theoretical Computer Science 192 (2007) 93–108
www.elsevier.com/locate/entcs

Reversibility and Models for Concurrency

Iain Phillips[1]
Department of Computing
Imperial College London
180 Queen's Gate, London, SW7 2AZ, United Kingdom

Irek Ulidowski[2]
Department of Computer Science
University of Leicester
University Road, Leicester, LE1 7RH, United Kingdom

tion law: $(a \mid (b+c)) + (a \mid b) + ((a+c) \mid b) = (a \mid (b+c)) + ((a+c) \mid b)$. We show that FR bisimulation coincides with *hereditary history-preserving (HHP) bisimulation*, which is regarded as the canonical true concurrency equivalence [1,5,7,3]. The result holds for reversible transition systems with no auto-concurrency and with no auto-causation, and since CCSK gives rise to such transition systems the result holds for CCSK.

Why do we care about HHPB?

ELSEVIER
Contents lists available at ScienceDirect
Journal of Logical and Algebraic Methods in Programming
www.elsevier.com/locate/jlamp

Contextual equivalences in configuration structures and reversibility ☆, ☆☆

CrossMark

Clément Aubert [a,b,*,1], Ioana Cristescu [c,**]

[a] INRIA, France
[b] Université Paris-Est, LACL (EA 4219), UPEC, F-94010 Créteil, France
[c] Diderot, Sorbonne Paris Cité, P.P.S., UMR 7126, F-75205 Paris, France

4. Conclusions and future work

We showed that, for a restricted class of RCCS processes (coherent, without recursion, auto-concurrency nor auto-conflict (Definition 26)) hereditary history preserving bisimilarity has a contextual characterization in CCS. We used the barbed congruence defined on RCCS as the congruence of reference, adapted it to configuration structures and then showed a

Clément Aubert, Ioana Cristescu

R_1 and R_2 in R are "simple" back-and-forth (SB&F) if

$\exists \mathcal{R} \subseteq \mathrm{R} \times \mathrm{R}$ such that

(Erroneous) Conjecture

R_1 and R_2 are SB&F iff $[\![R_1]\!]$ and $[\![R_2]\!]$ are HHPB.

Clément Aubert, Ioana Cristescu

R_1 and R_2 in R are "simple" back-and-forth (SB&F) if

$\exists \mathcal{R} \subseteq \mathrm{R} \times \mathrm{R}$ such that $(R_1, R_2) \in \mathcal{R}$,

(Erroneous) Conjecture

R_1 and R_2 are SB&F iff $[\![R_1]\!]$ and $[\![R_2]\!]$ are HHPB.

R_1 and R_2 in R are "simple" back-and-forth (SB&F) if

$\exists \mathcal{R} \subseteq \mathrm{R} \times \mathrm{R}$ such that $(R_1, R_2) \in \mathcal{R}$, and if R_i' is (forward or backward) reachable from R_i, $i \in \{1, 2\}$, and $(R_1', R_2') \in \mathcal{R}$, then

(Erroneous) Conjecture

R_1 and R_2 are SB&F iff $[\![R_1]\!]$ and $[\![R_2]\!]$ are HHPB.

R_1 and R_2 in R are "simple" back-and-forth (SB&F) if

$\exists \mathcal{R} \subseteq \mathrm{R} \times \mathrm{R}$ such that $(R_1, R_2) \in \mathcal{R}$, and if R_i' is (forward or backward) reachable from R_i, $i \in \{1, 2\}$, and $(R_1', R_2') \in \mathcal{R}$, then

$$\forall R_1' \xrightarrow{a} R_1'' \Rightarrow \exists R_2'', R_2' \xrightarrow{a} R_2''$$

$$\forall R_2' \xrightarrow{a} R_2'' \Rightarrow \exists R_1'', R_1' \xrightarrow{a} R_1''$$

(Erroneous) Conjecture

R_1 and R_2 are SB&F iff $[\![R_1]\!]$ and $[\![R_2]\!]$ are HHPB.

R_1 and R_2 in R are “simple” back-and-forth (SB&F) if

$\exists \mathcal{R} \subseteq \mathrm{R} \times \mathrm{R}$ such that $(R_1, R_2) \in \mathcal{R}$, and if R'_i is (forward or backward) reachable from R_i, $i \in \{1, 2\}$, and $(R'_1, R'_2) \in \mathcal{R}$, then

$$\forall R'_1 \xrightarrow{a} R''_1 \Rightarrow \exists R''_2, R'_2 \xrightarrow{a} R''_2$$

$$\forall R'_2 \xrightarrow{a} R''_2 \Rightarrow \exists R''_1, R'_1 \xrightarrow{a} R''_1$$

$$\forall R'_1 \overset{a}{\rightsquigarrow} R''_1 \Rightarrow \exists R''_2, R'_2 \overset{a}{\rightsquigarrow} R''_2$$

$$\forall R'_2 \overset{a}{\rightsquigarrow} R''_2 \Rightarrow \exists R''_1, R'_1 \overset{a}{\rightsquigarrow} R''_1$$

(Erroneous) Conjecture

R_1 and R_2 are SB&F iff $[\![R_1]\!]$ and $[\![R_2]\!]$ are HHPB.

The terms $a.a$ and $a \mid a$ are SB&F

$$a.a \xrightarrow{a} a \xrightarrow{a} 0$$

$$a \mid a \xrightarrow{a} a \xrightarrow{a} 0$$

The terms $a.a$ and $a \mid a$ are SB&F

$$a.a \xrightarrow{a} a \xrightarrow{a} 0$$

$$a \mid a \xrightarrow{a} a \xrightarrow{a} 0$$

The configurations $[\![a.a]\!]$ and $[\![a \mid a]\!]$ are *not* HHPB

$$\begin{tikzcd} \{a_1, a_2\} \\ \{a_1\} \arrow[u] \\ \emptyset \arrow[u] \end{tikzcd} \qquad \begin{tikzcd} & \{a_1, a_2\} & \\ \{a_1\} \arrow[ur] & & \{a_2\} \arrow[ul] \\ & \emptyset \arrow[ul] \arrow[ur] & \end{tikzcd}$$

$$\mathcal{R} = \{(\emptyset, \emptyset, \emptyset), \quad \}$$

How Reversibility Can Solve Traditional Questions: The Example of HHPB

The terms $a.a$ and $a \mid a$ are SB&F

$$a.a \xrightarrow{a} a \xrightarrow{a} 0$$

$$a \mid a \xrightarrow{a} a \xrightarrow{a} 0$$

The configurations $[\![a.a]\!]$ and $[\![a \mid a]\!]$ are *not* HHPB

$$\begin{tikzcd}
\{a_1, a_2\} \\
\{a_1\} \arrow[u] \\
\emptyset \arrow[u]
\end{tikzcd}
\qquad
\begin{tikzcd}
 & \{a_1, a_2\} & \\
\{a_1\} \arrow[ur] & & \{a_2\} \arrow[ul] \\
 & \emptyset \arrow[ul] \arrow[ur] &
\end{tikzcd}$$

$$\mathcal{R} = \{(\emptyset, \emptyset, \emptyset),$$

$$\}$$

The terms $a.a$ and $a \mid a$ are SB&F

$$\begin{aligned} a.a &\xrightarrow{a} a \xrightarrow{a} 0 \\ a \mid a &\xrightarrow{a} a \xrightarrow{a} 0 \end{aligned}$$

The configurations $[\![a.a]\!]$ and $[\![a \mid a]\!]$ are *not* HHPB

$$\begin{tikzcd} \{a_1, a_2\} \\ \boxed{\{a_1\}} \arrow[u] \\ \boxed{\varnothing} \arrow[u] \end{tikzcd} \qquad \begin{tikzcd} & \{a_1, a_2\} & \\ \boxed{\{a_1\}} \arrow[ur] & & \{a_2\} \arrow[ul] \\ & \boxed{\varnothing} \arrow[ul] \arrow[ur] & \end{tikzcd}$$

$$\mathcal{R} = \{(\varnothing, \varnothing, \varnothing), (\{a_1\}, \{a_1\}, \{a_1 \mapsto a_1\}),$$

$$\}$$

The terms $a.a$ and $a \mid a$ are SB&F

$$
\begin{aligned}
a.a &\xrightarrow{a} a \xrightarrow{a} 0\\
a \mid a &\xrightarrow{a} a \xrightarrow{a} 0
\end{aligned}
$$

The configurations $[\![a.a]\!]$ and $[\![a \mid a]\!]$ are *not* HHPB

$$
\begin{tikzcd}
\{a_1, a_2\} \\
\{a_1\} \arrow[u] \\
\varnothing \arrow[u]
\end{tikzcd}
\qquad
\begin{tikzcd}
 & \{a_1, a_2\} & \\
\{a_1\} \arrow[ur] & & \{a_2\} \arrow[ul] \\
 & \varnothing \arrow[ul] \arrow[ur] &
\end{tikzcd}
$$

$$
\mathcal{R} = \{(\varnothing, \varnothing, \varnothing), (\{a_1\}, \{a_1\}, \{a_1 \mapsto a_1\}),
\}
$$

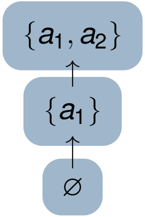
How Reversibility Can Solve Traditional Questions: The Example of HHPB

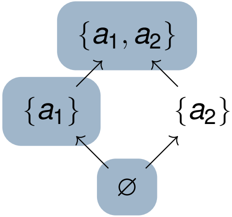
The terms $a.a$ and $a \mid a$ are SB&F

$$a.a \xrightarrow{a} a \xrightarrow{a} 0$$

$$a \mid a \xrightarrow{a} a \xrightarrow{a} 0$$

The configurations $[\![a.a]\!]$ and $[\![a \mid a]\!]$ are *not* HHPB





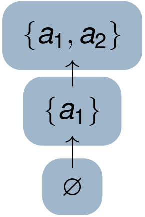
$$\mathcal{R} = \{(\emptyset, \emptyset, \emptyset), (\{a_1\}, \{a_1\}, \{a_1 \mapsto a_1\}),$$
$$(\{a_1, a_2\}, \{a_1, a_2\}, \{a_1 \mapsto a_1, a_2 \mapsto a_2\}),$$
$$\}$$

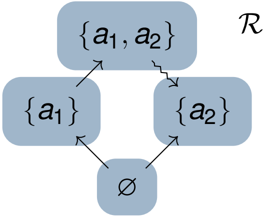
The terms $a.a$ and $a \mid a$ are SB&F

$$a.a \xrightarrow{a} a \xrightarrow{a} 0$$

$$a \mid a \xrightarrow{a} a \xrightarrow{a} 0$$

The configurations $[\![a.a]\!]$ and $[\![a \mid a]\!]$ are *not* HHPB





$$\begin{aligned}\mathcal{R} = \{&(\varnothing, \varnothing, \varnothing), (\{a_1\}, \{a_1\}, \{a_1 \mapsto a_1\}),\\ &(\{a_1, a_2\}, \{a_1, a_2\}, \{a_1 \mapsto a_1, a_2 \mapsto a_2\}),\\ \\ \}\end{aligned}$$

The terms $a.a$ and $a \mid a$ are SB&F

$$a.a \xrightarrow{a} a \xrightarrow{a} 0$$

$$a \mid a \xrightarrow{a} a \xrightarrow{a} 0$$

The configurations $[\![a.a]\!]$ and $[\![a \mid a]\!]$ are *not* HHPB

$$\begin{tikzcd} \{a_1, a_2\} \arrow[d, squiggly] \\ \{a_1\} \\ \varnothing \arrow[u] \end{tikzcd}$$

$$\begin{tikzcd} & \{a_1, a_2\} \arrow[dr, squiggly] & \\ \{a_1\} \arrow[ur] & & \{a_2\} \\ & \varnothing \arrow[ul] \arrow[ur] & \end{tikzcd}$$

$$\begin{aligned} \mathcal{R} = \{ & (\varnothing, \varnothing, \varnothing), (\{a_1\}, \{a_1\}, \{a_1 \mapsto a_1\}), \\ & (\{a_1, a_2\}, \{a_1, a_2\}, \{a_1 \mapsto a_1, a_2 \mapsto a_2\}), \\ & (\{a_1\}, \{a_2\}, \{a_1 \mapsto a_2\}), \cdots \\ \} & \end{aligned}$$

Clément Aubert, Ioana Cristescu

What we know so far

— The “right” equivalence for reversible calculi is *not* “just” back-and-forth + labels,

What we know so far

- The “right” equivalence for reversible calculi is *not* “just” back-and-forth + labels,
- Two transitions with the same label cannot be distinguished (auto-concurrency),

What we know so far

- The “right” equivalence for reversible calculi is *not* “just” back-and-forth + labels,
- Two transitions with the same label cannot be distinguished (auto-concurrency),

$\Rightarrow$ Use RCCS’ identifiers!

What we know so far

- The “right” equivalence for reversible calculi is *not* “just” back-and-forth + labels,
- Two transitions with the same label cannot be distinguished (auto-concurrency),

$\Rightarrow$ Use RCCS’ identifiers!

$$R \overset{i:a}{\rightsquigarrow} \cdots \overset{i_n:a_n}{\rightsquigarrow} O_R$$

RCCS' identifiers

Reversible Communicating Systems

Vincent Danos[1]⋆ and Jean Krivine[2]

[1] Université Paris 7 & CNRS
[2] INRIA Rocquencourt

Monitored Processes. In RCCS, simple processes are not runnable as such, only monitored processes are. This second kind of process is defined as follows:

Memories:	$m ::= \langle\rangle$	Empty memory
	$\mid \langle 1 \rangle \cdot m$	Left-Fork
	$\mid \langle 2 \rangle \cdot m$	Right-Fork
	$\mid \langle *, \alpha, P \rangle \cdot m$	Semi-Synch
	$\mid \langle m, \alpha, P \rangle \cdot m$	Synch
Monitored Processes:	$R ::= m \triangleright P$	Threads
	$\mid (R \mid R)$	Product
	$\mid (a)R$	Restriction

RCCS' identifiers

Contents lists available at ScienceDirect
Journal of Logical and Algebraic Methods in Programming
www.elsevier.com/locate/jlamp

Contextual equivalences in configuration structures and reversibility[☆,☆☆]

Clément Aubert[a,b,*,1], Ioana Cristescu[c,**]

[a] INRIA, France
[b] Université Paris-Est, LACL (EA 4219), UPEC, F-94010 Créteil, France
[c] Diderot, Sorbonne Paris Cité, P.P.S., UMR 7126, F-75205 Paris, France

Grammar. Consider the following *process constructors*, also called *combinators* or *operators*:

$$e := \langle i, \alpha, P \rangle \qquad \text{(memory events)}$$

$$m := \emptyset \;\|\; \curlyvee.m \;\|\; e.m \qquad \text{(memory stacks)}$$

$$P, Q := \lambda.P \;\|\; P \mid Q \;\|\; \lambda.P + \pi.Q \;\|\; P\backslash a \;\|\; 0 \qquad \text{(CCS processes)}$$

$$R, S := m \rhd P \;\|\; R \mid S \;\|\; R\backslash a \qquad \text{(RCCS processes)}$$

A (memory) event $e = \langle i, \alpha, P \rangle$ is made of:

- An event identifier $i \in \mathsf{I}$ that *tags* transitions. We may think of them as `pid`, in the sense that they are a centrally distributed identifier attached to each transition.

RCCS' identifiers

Contents lists available at ScienceDirect
Journal of Logical and Algebraic Methods in Programming
www.elsevier.com/locate/jlamp

Contextual equivalences in configuration structures and reversibility [☆,☆☆]

Clément Aubert [a,b,*,1], Ioana Cristescu [c,**]

[a] INRIA, France
[b] Université Paris-Est, LACL (EA 4219), UPEC, F-94010 Créteil, France
[c] Diderot, Sorbonne Paris Cité, P.P.S., UMR 7126, F-75205 Paris, France

$$\textsc{In}+\ \frac{}{m \rhd a.P + Q \xrightarrow{i:a} \langle i, a, Q \rangle.m \rhd P}\ i \notin \mathsf{I}(m)$$

$$\textsc{Out}+\ \frac{}{m \rhd \bar{a}.P + Q \xrightarrow{i:\bar{a}} \langle i, \bar{a}, Q \rangle.m \rhd P}\ i \notin \mathsf{I}(m)$$

$$\textsc{In}-\ \frac{}{\langle i, a, Q \rangle.m \rhd P \overset{i:a}{\rightsquigarrow} m \rhd a.P + Q}\ i \notin \mathsf{I}(m)$$

$$\textsc{Out}-\ \frac{}{\langle i, \bar{a}, Q \rangle.m \rhd P \overset{i:\bar{a}}{\rightsquigarrow} m \rhd \bar{a}.P + Q}\ i \notin \mathsf{I}(m)$$

(a) Prefix and sum rules

$$\textsc{Com}+\ \frac{R \xrightarrow{i:\alpha} R' \quad S \xrightarrow{i:\bar{\alpha}} S'}{R \mid S \xrightarrow{i:\tau} R' \mid S'} \qquad \textsc{ParL}\ \frac{R \overset{i:\alpha}{\twoheadrightarrow} R'}{R \mid S \overset{i:\alpha}{\twoheadrightarrow} R' \mid S}\ i \notin \mathsf{I}(S)$$

$$\textsc{Com}-\ \frac{R \overset{i:\alpha}{\rightsquigarrow} R' \quad S \overset{i:\bar{\alpha}}{\rightsquigarrow} S'}{R \mid S \overset{i:\tau}{\rightsquigarrow} R' \mid S'} \qquad \textsc{ParR}\ \frac{R \overset{i:\alpha}{\twoheadrightarrow} R'}{S \mid R \overset{i:\alpha}{\twoheadrightarrow} S \mid R'}\ i \notin \mathsf{I}(S)$$

(b) Parallel constructions

How Reversibility Can Solve Traditional Questions: The Example of HHPB

$\mathcal{C}_1$ and $\mathcal{C}_2$ are HHPB if

$\exists \mathcal{R} \subseteq C_1 \times C_2 \times (E_1 \rightharpoonup E_2)$ such that $(\varnothing, \varnothing, \varnothing) \in \mathcal{R}$, and if $(x_1, x_2, f) \in \mathcal{R}$, then f is a label- and order- preserving bijection between x_1 and x_2 such that:

$$\forall y_1, x_1 \xrightarrow{e_1} y_1 \Rightarrow \exists y_2, g, x_2 \xrightarrow{e_2} y_2, g\restriction_{x_1} = f, (y_1, y_2, g) \in \mathcal{R}$$

$$\forall y_2, x_2 \xrightarrow{e_2} y_2 \Rightarrow \exists y_1, g, x_1 \xrightarrow{e_1} y_1, g\restriction_{x_1} = f, (y_1, y_2, g) \in \mathcal{R}$$

$$\forall y_1, x_1 \overset{e_1}{\rightsquigarrow} y_1 \Rightarrow \exists y_2, g, x_2 \overset{e_2}{\rightsquigarrow} y_2, g = f\restriction_{y_1}, (y_1, y_2, g) \in \mathcal{R}$$

$$\forall y_2, x_2 \overset{e_2}{\rightsquigarrow} y_2 \Rightarrow \exists y_1, g, x_1 \overset{e_1}{\rightsquigarrow} y_1, g = f\restriction_{y_1}, (y_1, y_2, g) \in \mathcal{R}$$

R_1 and R_2 are B&F if

$\exists \mathcal{R} \subseteq C_1 \times C_2 \times (E_1 \rightharpoonup E_2)$ such that $(\emptyset, \emptyset, \emptyset) \in \mathcal{R}$, and if $(x_1, x_2, f) \in \mathcal{R}$, then f is a label- and order- preserving bijection between x_1 and x_2 such that:

$$\forall y_1, x_1 \xrightarrow{e_1} y_1 \Rightarrow \exists y_2, g, x_2 \xrightarrow{e_2} y_2, g\restriction_{x_1} = f, (y_1, y_2, g) \in \mathcal{R}$$

$$\forall y_2, x_2 \xrightarrow{e_2} y_2 \Rightarrow \exists y_1, g, x_1 \xrightarrow{e_1} y_1, g\restriction_{x_1} = f, (y_1, y_2, g) \in \mathcal{R}$$

$$\forall y_1, x_1 \overset{e_1}{\rightsquigarrow} y_1 \Rightarrow \exists y_2, g, x_2 \overset{e_2}{\rightsquigarrow} y_2, g = f\restriction_{y_1}, (y_1, y_2, g) \in \mathcal{R}$$

$$\forall y_2, x_2 \overset{e_2}{\rightsquigarrow} y_2 \Rightarrow \exists y_1, g, x_1 \overset{e_1}{\rightsquigarrow} y_1, g = f\restriction_{y_1}, (y_1, y_2, g) \in \mathcal{R}$$

Clément Aubert, Ioana Cristescu

How Reversibility Can Solve Traditional Questions: The Example of HHPB

R_1 and R_2 are B&F if

$\exists \mathcal{R} \subseteq R \times R \times I \rightharpoonup I$ such that $(\varnothing \triangleright O_{R_1}, \varnothing \triangleright O_{R_2}, \varnothing) \in \mathcal{R}$, and if $(x_1, x_2, f) \in \mathcal{R}$, then f is a label- and order- preserving bijection between x_1 and x_2 such that:

$$\forall y_1, x_1 \xrightarrow{e_1} y_1 \Rightarrow \exists y_2, g, x_2 \xrightarrow{e_2} y_2, g{\restriction_{x_1}} = f, (y_1, y_2, g) \in \mathcal{R}$$

$$\forall y_2, x_2 \xrightarrow{e_2} y_2 \Rightarrow \exists y_1, g, x_1 \xrightarrow{e_1} y_1, g{\restriction_{x_1}} = f, (y_1, y_2, g) \in \mathcal{R}$$

$$\forall y_1, x_1 \overset{e_1}{\rightsquigarrow} y_1 \Rightarrow \exists y_2, g, x_2 \overset{e_2}{\rightsquigarrow} y_2, g = f{\restriction_{y_1}}, (y_1, y_2, g) \in \mathcal{R}$$

$$\forall y_2, x_2 \overset{e_2}{\rightsquigarrow} y_2 \Rightarrow \exists y_1, g, x_1 \overset{e_1}{\rightsquigarrow} y_1, g = f{\restriction_{y_1}}, (y_1, y_2, g) \in \mathcal{R}$$

R_1 and R_2 are B&F if

$\exists \mathcal{R} \subseteq R \times R \times I \rightharpoonup I$ such that $(\emptyset \triangleright O_{R_1}, \emptyset \triangleright O_{R_2}, \emptyset) \in \mathcal{R}$, and if $(R_1, R_2, f) \in \mathcal{R}$, then f is a ~~label- and order- preserving~~ bijection between $I(R_1)$ and $I(R_2)$ such that:

$$\forall y_1, x_1 \xrightarrow{e_1} y_1 \Rightarrow \exists y_2, g, x_2 \xrightarrow{e_2} y_2, g\restriction_{x_1} = f, (y_1, y_2, g) \in \mathcal{R}$$

$$\forall y_2, x_2 \xrightarrow{e_2} y_2 \Rightarrow \exists y_1, g, x_1 \xrightarrow{e_1} y_1, g\restriction_{x_1} = f, (y_1, y_2, g) \in \mathcal{R}$$

$$\forall y_1, x_1 \overset{e_1}{\rightsquigarrow} y_1 \Rightarrow \exists y_2, g, x_2 \overset{e_2}{\rightsquigarrow} y_2, g = f\restriction_{y_1}, (y_1, y_2, g) \in \mathcal{R}$$

$$\forall y_2, x_2 \overset{e_2}{\rightsquigarrow} y_2 \Rightarrow \exists y_1, g, x_1 \overset{e_1}{\rightsquigarrow} y_1, g = f\restriction_{y_1}, (y_1, y_2, g) \in \mathcal{R}$$

How Reversibility Can Solve Traditional Questions: The Example of HHPB

R_1 and R_2 are B&F if

$\exists \mathcal{R} \subseteq R \times R \times I \rightharpoonup I$ such that $(\emptyset \triangleright O_{R_1}, \emptyset \triangleright O_{R_2}, \emptyset) \in \mathcal{R}$, and if $(R_1, R_2, f) \in \mathcal{R}$, then f is a ~~label- and order- preserving~~ bijection between $I(R_1)$ and $I(R_2)$ such that:

$$\forall R_1', R_1 \xrightarrow{e_1} R_1' \Rightarrow \exists R_2', g, R_2 \xrightarrow{e_2} R_2', g\restriction_{R_1} = f, (R_1', R_2', g) \in \mathcal{R}$$

$$\forall R_2', R_2 \xrightarrow{e_2} R_2' \Rightarrow \exists R_1', g, R_1 \xrightarrow{e_1} R_1', g\restriction_{R_1} = f, (R_1', R_2', g) \in \mathcal{R}$$

$$\forall R_1', R_1 \overset{e_1}{\rightsquigarrow} R_1' \Rightarrow \exists R_2', g, R_2 \overset{e_2}{\rightsquigarrow} R_2', g = f\restriction_{R_1'}, (R_1', R_2', g) \in \mathcal{R}$$

$$\forall R_2', R_2 \overset{e_2}{\rightsquigarrow} R_2' \Rightarrow \exists R_1', g, R_1 \overset{e_1}{\rightsquigarrow} R_1', g = f\restriction_{R_1'}, (R_1', R_2', g) \in \mathcal{R}$$

R_1 and R_2 are B&F if

$\exists \mathcal{R} \subseteq R \times R \times I \rightharpoonup I$ such that $(\varnothing \triangleright O_{R_1}, \varnothing \triangleright O_{R_2}, \varnothing) \in \mathcal{R}$, and if $(R_1, R_2, f) \in \mathcal{R}$, then f is a ~~label- and order- preserving~~ bijection between $I(R_1)$ and $I(R_2)$ such that:

$$\forall R_1', R_1 \xrightarrow{i_1:a} R_1' \Rightarrow \exists R_2', g, R_2 \xrightarrow{i_2:a} R_2', g\restriction_{R_1} = f, (R_1', R_2', g) \in \mathcal{R}$$

$$\forall R_2', R_2 \xrightarrow{i_2:a} R_2' \Rightarrow \exists R_1', g, R_1 \xrightarrow{i_1:a} R_1', g\restriction_{R_1} = f, (R_1', R_2', g) \in \mathcal{R}$$

$$\forall R_1', R_1 \overset{i_1:a}{\rightsquigarrow} R_1' \Rightarrow \exists R_2', g, R_2 \overset{i_2:a}{\rightsquigarrow} R_2', g = f\restriction_{R_1'}, (R_1', R_2', g) \in \mathcal{R}$$

$$\forall R_2', R_2 \overset{i_2:a}{\rightsquigarrow} R_2' \Rightarrow \exists R_1', g, R_1 \overset{i_1:a}{\rightsquigarrow} R_1', g = f\restriction_{R_1'}, (R_1', R_2', g) \in \mathcal{R}$$

R_1 and R_2 are B&F if

$\exists \mathcal{R} \subseteq R \times R \times I \rightharpoonup I$ such that $(\emptyset \triangleright O_{R_1}, \emptyset \triangleright O_{R_2}, \emptyset) \in \mathcal{R}$, and if $(R_1, R_2, f) \in \mathcal{R}$, then f is a ~~label- and order- preserving~~ bijection between $I(R_1)$ and $I(R_2)$ such that:

$$\forall R_1', R_1 \xrightarrow{i_1:a} R_1' \Rightarrow \exists R_2', g, R_2 \xrightarrow{i_2:a} R_2', g\restriction_{R_1} = f, (R_1', R_2', g) \in \mathcal{R}$$

$$\forall R_2', R_2 \xrightarrow{i_2:a} R_2' \Rightarrow \exists R_1', g, R_1 \xrightarrow{i_1:a} R_1', g\restriction_{R_1} = f, (R_1', R_2', g) \in \mathcal{R}$$

$$\forall R_1', R_1 \overset{i_1:a}{\rightsquigarrow} R_1' \Rightarrow \exists R_2', g, R_2 \overset{i_2:a}{\rightsquigarrow} R_2', g = f\restriction_{R_1'}, (R_1', R_2', g) \in \mathcal{R}$$

$$\forall R_2', R_2 \overset{i_2:a}{\rightsquigarrow} R_2' \Rightarrow \exists R_1', g, R_1 \overset{i_1:a}{\rightsquigarrow} R_1', g = f\restriction_{R_1'}, (R_1', R_2', g) \in \mathcal{R}$$

- f preserves labels “for free”,
- f will always have to (un-)match i_1 and i_2,
- identifiers *will* induce an order on the transitions.

Main result

R_1 and R_2 are B&F iff $[\![O_{R_1}]\!]$ and $[\![O_{R_2}]\!]$ are HHPB.

Main result

P_1 and P_2 are B&F iff $[\![P_1]\!]$ and $[\![P_2]\!]$ are HHPB.

Main result

P_1 and P_2 are B&F iff $[\![P_1]\!]$ and $[\![P_2]\!]$ are HHPB.

By-product

On processes without auto-concurrency, B&F = SB&F.

Main result

P_1 and P_2 are B&F iff $[\![P_1]\!]$ and $[\![P_2]\!]$ are HHPB.

By-product

On processes without auto-concurrency, B&F = SB&F.

Techniques

— Encoding of memory,

— Categorical representation,

— Operational correspondence with new model,

— Trace equivalences,

— Connection to previous semantics of reversible calculi.

Main result

P_1 and P_2 are B&F iff $[\![P_1]\!]$ and $[\![P_2]\!]$ are HHPB.

By-product

On processes without auto-concurrency, B&F = SB&F.

Techniques

- Encoding of memory,
- Categorical representation,
- Operational correspondence with new model,
- Trace equivalences,
- Connection to previous semantics of reversible calculi.

Example of memory encoding and its correspondence

$$
[\![P]\!] = \begin{tikzcd}[column sep=small]
& \{a_1, a_2\} & \\
\{a_1\} \arrow[ur] & & \{a_2\} \arrow[ul] \\
& \varnothing \arrow[ul] \arrow[ur] &
\end{tikzcd}
\qquad
P = a \mid a
$$

Example of memory encoding and its correspondence

$$
\lfloor \emptyset \rfloor = \begin{tikzcd}[column sep=small]
 & \{a_1, a_2\} & \\
\{a_1\} \arrow[ur] & & \{a_2\} \arrow[ul] \\
 & \emptyset \arrow[ul] \arrow[ur] &
\end{tikzcd}
\qquad
\emptyset \triangleright P \equiv \curlyvee.\emptyset \triangleright a \mid \curlyvee.\emptyset \triangleright a
$$

Clément Aubert, Ioana Cristescu

Example of memory encoding and its correspondence

$$\begin{tikzcd}[column sep=small]
 & \{a_1^1, a_2\} & \\
\boxed{\{a_1^1\}} \arrow[ur] & & \{a_2\} \arrow[ul] \\
 & \boxed{\varnothing} \arrow[ul] \arrow[ur] &
\end{tikzcd}$$

$$\lfloor \langle 1, a\rangle . \curlyvee . \varnothing \rfloor = \boxed{\varnothing}$$

$$\boxed{\varnothing \rhd} P \equiv \boxed{\curlyvee.\varnothing \rhd} a \mid \boxed{\curlyvee.\varnothing \rhd} a$$

$$\xrightarrow{1:a} \boxed{\langle 1, a\rangle . \curlyvee . \varnothing} \rhd 0 \mid \boxed{\curlyvee.\varnothing} \rhd a$$

In a nutshell (again!)

We solved
an open problem
using
reversibility
and offering
a new model.

In a nutshell (again!)

We solved
the question of the capturing HHPB in syntactical terms
using
reversibility
and offering
a new model.

In a nutshell (again!)

We solved
the question of the capturing HHPB in syntactical terms
using
back-and-forth transitions *and* the memory mechanism
and offering
a new model.

In a nutshell (again!)

We solved
the question of the capturing HHPB in syntactical terms
using
back-and-forth transitions *and* the memory mechanism
and offering
identified configuration structures.

In a nutshell (again!)

We solved
the question of the capturing HHPB in syntactical terms
using
back-and-forth transitions *and* the memory mechanism
and offering
identified configuration structures.

Thanks!

We'll be in the chat to answer your questions!