Towards a Categorical Representation of Reversible Event Structures

Eva Graversen Iain Phillips Nobuko Yoshida
Imperial College London, UK

We study categories for reversible computing, focussing on reversible forms of event structures. Event structures are a well-established model of true concurrency. There exist a number of forms of event structures, including prime event structures, asymmetric event structures, and general event structures. More recently, reversible forms of these types of event structures have been defined. We formulate corresponding categories and functors between them. We show that products and coproducts exist in many cases. In most work on reversible computing, including reversible process calculi, a cause-respecting condition is posited, meaning that the cause of an event may not be reversed before the event itself. Since reversible event structures are not assumed to be cause-respecting in general, we also define cause-respecting subcategories of these event structures. Our longer-term aim is to formulate event structure semantics for reversible process calculi.

1 Introduction

Event structures [10], a well-known model of true concurrency, consist of events and relations between them, describing the causes of events and conflict between events. Winskel [18] defined a category of event structures, and used this to define event structure semantics of CCS.

Reversible process calculi are a well-studied field [3, 5, 6, 8, 9, 11]. When considering the semantics of reversible processes, the ability to reverse events leads to finer distinctions of a true concurrency character [12]; for example the CCS processes $a \mid b$ and $a.b + b.a$ can easily be distinguished by whether both a and b can be reversed at the end of the computation. This motivates the study of *reversible event structures*. So far, no event structure semantics have been defined for reversible variants of CCS [5, 6, 11] (though the reversible π-calculus has been modelled using rigid families [4]); we intend this work to be one of the first steps towards doing so.

Reversible versions of various kinds of event structures were introduced in [13, 15]. Our aim here is to interpret these as objects in appropriate categories and study functors between them. So far few reversible frameworks have been defined categorically, though [7] used category theory to describe the relationship between RCCS processes and their histories, and [2] used dagger categories to define a reversible process calculus called Π.

We define categories for the reversible event structures from [13, 15], defining morphisms for each category and functors, and in some cases adjunctions, between them, along with coproducts, and, in the case of general reversible event structures, products. We also consider configuration systems [13], a model of concurrency intended to serve a similar purpose as domains do for the forward-only event structures, letting the various kinds of reversible event structures be translated into one formalism. We show that, just as stable domains can be modelled as event structures, so finitely enabled configuration systems can be modelled as general reversible event structures, giving a tight correspondence (Theorem 5.9).

With a few exceptions [14, 16], reversible process calculi have always adopted *causal* reversibility. The reversible event structures of [13, 15] allow non-causal reversibility, inspired by bonding in biochemical processes. We here define subcategories of the reversible event structures of [15] which are

Submitted to:
PLACES 2017

$$\begin{tikzcd}
\mathbf{PES} \arrow[r, bend left, "A"] \arrow[r, phantom, "\top"] \arrow[rr, bend left=40, "P_{ps}"] \arrow[dr, bend left=10, "D_p"] & \mathbf{AES} \arrow[l, bend left, "\Sigma"] \arrow[d, "D_a"] & \mathbf{SES} \arrow[r, "I"] \arrow[dl, "D_s"] & \mathbf{ES} \\
& \mathbf{Dom} \arrow[ul, bend left=10, "P_{pd}"] & &
\end{tikzcd}$$

Figure 1: Categories of forward-only event structures and functors between them: PES were introduced in [10], and defined categorically along with **Dom**, **SES**, **ES**, D_p, P_{pd}, P_{ps}, and D_s in [19], **AES**, A, and D_a were introduced in [1], and Σ in [13]. The adjunction between A and Σ, denoted by $\dashv$, is new.

(1) *stable*, meaning that the causes of an event cannot be ambiguous, which is clearly important for reversibility, and (2) *cause-respecting*, meaning that no action can be reversed unless all the actions caused by it have been reversed first [13], which can be seen as a safety property for causal reversibility. We show that under these conditions any reachable configuration is forwards reachable (Theorem 6.8).

Structure of the Paper. Section 2 reviews forwards-only event structures; Section 3 looks at reversible prime and asymmetric event structures, while Section 4 covers reversible general event structures. Section 5 describes the category of configuration systems, and Section 6 describes stable and cause-respecting reversible event structures and configuration systems.

2 Forwards-Only Event Structures

Before describing the different categories of reversible event structures, we recall the categories of forward-only event structures and functors between them, as seen in Figure 1.

A *prime event structure* consists of a set of events, and causality and conflict relations describing when these events can occur. If $e < e'$ then e' cannot happen unless e has already happened. And if $e \sharp e'$ then e and e' each prevent each other from occurring.

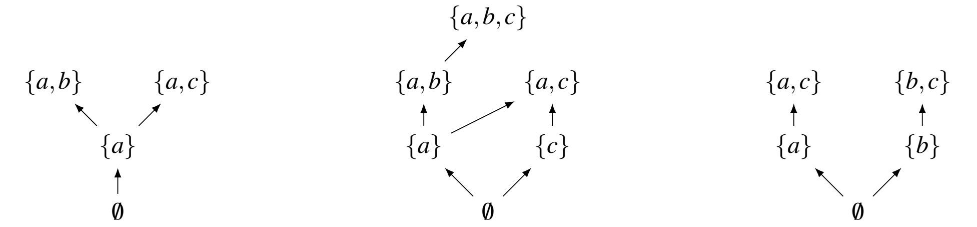
Definition 2.1 (Prime Event Structure [10])**.** *A prime event structure (PES) is a triple $\mathscr{E} = (E, <, \sharp)$, where E is the set of events and* causality*, $<$, and* conflict*, $\sharp$, are binary relations on E such that $\sharp$ is irreflexive and symmetric, $<$ is an irreflexive partial order such that for every $e \in E$, $\{e' \mid e' < e\}$ is finite, and $\sharp$ is hereditary with respect to $<$, i.e. for all $e, e', e'' \in E$, if $e \sharp e'$ and $e < e''$ then $e'' \sharp e'$.*

For any PES $\mathscr{E} = (E, <, \sharp)$, we say that $X \subseteq E$ is a configuration of $\mathscr{E}$ if X is left-closed under $<$ and conflict-free, meaning no $e, e' \in X$ exist, such that $e \sharp e'$. Configurations can be ordered by inclusion to form stable domains (coherent, prime algebraic, finitary partial orders) [19], as seen in Example 2.2.

Example 2.2. *The PES $\mathscr{E}_1$ with events a, b, c where $a < b$, $a < c$, and $c \sharp b$, has configurations $\emptyset$, $\{a\}$, $\{a, b\}$, and $\{a, c\}$, forming the domain seen in Figure 2a.*

Morphisms are defined on PESs in Definition 2.3, yielding the category **PES**. Morphisms on event structures act as a sort of synchronisation between the two structures, where if X is a configuration then $f(X)$ is too, and two events, e, e' can only synchronise with the same $f(e) = f(e')$ if they are in conflict.

Definition 2.3 (PES morphism [19])**.** *Let $\mathscr{E}_0 = (E_0, <_0, \sharp_0)$ and $\mathscr{E}_1 = (E_1, <_1, \sharp_1)$ be PESs. A morphism $f : \mathscr{E}_0 \to \mathscr{E}_1$ is a partial function $f : E_0 \to E_1$ such that for all $e \in E_0$, if $f(e) \neq \bot$ then $\{e_1 \mid e_1 <_1 f(e)\} \subseteq \{f(e') \mid e' <_0 e\}$, and for all $e, e' \in E_0$, if $f(e) \neq \bot \neq f(e')$ and $f(e) \sharp_1 f(e')$ or $f(e) = f(e')$ then $e \sharp_0 e'$ or $e = e'$.*



(a) Domain of PES $\mathscr{E}_1$ (b) Domain of AES $\mathscr{E}_2$ (c) Domain of SES $\mathscr{E}_3$

Figure 2: Examples of domains representing event structures.

Asymmetric event structures [1] resemble prime event structures, with the difference being that the conflict relation $e \rhd e'$ ([1] uses the notation $e \nearrow e'$) is asymmetric, so that rather than e and e' being unable to coexist in a configuration, e' cannot be added to a configuration that contains e. The converse relation $e' \lhd e$ can be seen as precedence or weak causation, where if both events are in a configuration then e' was added first, as illustrated by Example 2.4. An AES-morphism is defined in the same way as a PES morphism, but replacing symmetric conflict with asymmetric. This gives the category **AES**.

Example 2.4. *$\mathscr{E}_2 = (E,<,\lhd)$ where $E = \{a,b,c\}$ and $a < b$ and $b \lhd c$ has configurations $\emptyset$, $\{a\}$, $\{c\}$, $\{a,b\}$, $\{a,c\}$, and $\{a,b,c\}$, and therefore $D_a(\mathscr{E}_2)$ is the domain seen in Figure 2b.*

General event structures, or simply *event structures*, work somewhat differently from PESs or AESs. Instead of causation and conflict, they have an enabling relation and a consistency relation.

Definition 2.5 (Event structure [19])**.** *An event structure (ES) is a triple $\mathscr{E} = (E, \mathit{Con}, \vdash)$, where E is a set of events, $\mathit{Con} \subseteq_{\text{fin}} 2^E$ is the consistency relation, such that if $X \in \mathit{Con}$ and $Y \subseteq X$ then $Y \in \mathit{Con}$, and $\vdash \subseteq \mathit{Con} \times E$ is the enabling relation, such that if $X \vdash e$ and $X \subseteq Y \in \mathit{Con}$ then $Y \vdash e$.*

Configurations are finitely consistent sets of events, where each event is deducible via the enabling relation. Once again we define an ES-morphism, giving us the category **ES** [19]. The idea behind them is much the same as for PES- and AES-morphisms. Enabling sets are treated in much the same way as causes, and consistent sets in the opposite way from conflict.

Stable event structures [19] form a full subcategory **SES** of **ES**. The idea is that in any given configuration, each event will have a unique enabling set.

Example 2.6. *$\mathscr{E}_3 = (E, \mathit{Con}, \vdash)$ where $E = \{a,b,c\}$, $\mathit{Con} = \{\emptyset, \{a\}, \{b\}, \{a,c\}, \{b,c\}\}$, and $\emptyset \vdash a$, $\emptyset \vdash b$, $\{a\} \vdash c$, and $\{b\} \vdash c$ can be represented by the domain $D_s(\mathscr{E}_3)$ seen in Figure 2c.*

3 Reversible Prime and Asymmetric Event Structures

Our goal is to define the categories and functors in the lower part of Figure 3.

We start by adding reversibility to PESs. When discussing reversible events we will use $\underline{e}$ to denote reversing e and e^* to denote that e may be performed or reversed. *Reversible prime event structures* [13] (Definition 3.1) consist of a set of events, E, some of which may be reversible, causality and conflict similar to a PES, reverse causality, which works similarly to causality, in that $e \prec \underline{e'}$ means e' can only be reversed in configurations containing e, and prevention, which resembles the asymmetric conflict of AESs, in that $e \rhd \underline{e'}$ means that e' can only be reversed in configurations not containing e.

Definition 3.1 (RPES [13])**.** *A reversible prime event structure (RPES) is a sextuple $\mathscr{E} = (E,F,<,\sharp,\prec,\rhd)$ where E is the set of events, $F \subseteq E$ is the set of reversible events, and*

$$\begin{tikzcd}
\mathbf{PES} \arrow[rrr, bend left=30, "P_{ps}"] \arrow[rr, bend left=10, "A"] \arrow[rr, phantom, "\top"] \arrow[d, bend right, "P_p"'] \arrow[d, phantom, "\dashv"] & & \mathbf{AES} \arrow[ll, bend left=10, "\Sigma"] \arrow[d, bend right, "P_a"'] \arrow[d, phantom, "\dashv"] & \mathbf{SES} \arrow[r, "I"] \arrow[d, "P_r"] & \mathbf{ES} \arrow[d, "P_r"] & \\
\mathbf{RPES} \arrow[rr, "A_r"] \arrow[u, bend right, "\Phi_p"'] \arrow[dr, "C_{pr}"'] & & \mathbf{RAES} \arrow[r, "P_{ar}"] \arrow[u, bend right, "\Phi_a"'] \arrow[dl, "C_{ar}"'] & \mathbf{SRES} \arrow[r, "I"] \arrow[dll, bend right=10, "C_r"'] \arrow[dll, phantom, "\perp"] & \mathbf{RES} \arrow[d, bend right, "C_r"'] \arrow[d, phantom, "\dashv"] & \\
& \mathbf{SCS} \arrow[rrr, "I"] \arrow[urr, bend right=10, "R"'] & & & \mathbf{FCS} \arrow[r, "I"'] \arrow[u, bend right, "R"'] & \mathbf{CS}
\end{tikzcd}$$

Figure 3: Categories of event structures and functors between them: We extend Figure 1 by categorically defining RPESs, RAESs, CSs, P_p, Φ_p, P_a, Φ_a, C_p, C_{pr}, C_a, C_{ar}, and A_r [13] and RESs and P_r [15]. The categories **SRES**, **SCS**, and **FCS**, and functors P_{pr}, P_{ar}, C_r, C, and R are new, as well as the noted adjunctions.

- *$<$ is an irreflexive partial order such that for every $e \in E$, $\{e' \in E \mid e' < e\}$ is finite and conflict-free*
- *$\sharp$ is irreflexive and symmetric such that if $e < e'$ then not $e \,\sharp\, e'$*
- *$\rhd \subseteq E \times \underline{F}$ is the prevention relation*
- *$\prec \subseteq E \times \underline{F}$ is the reverse causality relation where for each $e \in F$, $e \prec \underline{e}$ and $\{e' \mid e' \prec \underline{e}\}$ is finite and conflict-free and if $e \prec \underline{e'}$ then not $e \rhd \underline{e'}$*
- *$\sharp$ is hereditary with respect to sustained causation $\ll$ and $\ll$ is transitive, where $e \ll e'$ means that $e < e'$ and if $e \in F$ then $e' \rhd \underline{e}$*

As previously, in order to define the category **RPES**, we need a notion of morphism. An RPES-morphism can be seen as a combination of a PES-morphism for the forwards part and an AES-morphism for the reverse part, and reversible events can only synchronise with other reversible events. The category **RPES** has coproducts (Definition 3.2). A coproduct can be described as a choice between two event structures to behave as, as illustrated by Example 3.3.

Definition 3.2 (RPES coproduct)**.** *Given RPESs $\mathscr{E}_0 = (E_0, F_0, <_0, \sharp_0, \prec_0, \rhd_0)$ and $\mathscr{E}_1 = (E_1, F_0, <_1, \sharp_1, \prec_1, \rhd_1)$, their coproduct $\mathscr{E}_0 + \mathscr{E}_1$ is $(E, F, <, \sharp, \prec, \rhd)$ where:*

- *$E = \{(0,e) \mid e \in E_0\} \cup \{(1,e) \mid e \in E_1\}$ and $F = \{(0,e) \mid e \in F_0\} \cup \{(1,e) \mid e \in F_1\}$*
- *injection i_j exist such that for $e \in E_j$, $i_j(e) = (j,e)$ for $j \in \{0,1\}$*
- *$(j,e) < (j',e')$ iff $j = j'$ and $e <_j e'$*
- *$(j,e) \,\sharp\, (j',e')$ iff $j \neq j'$ or $e \,\sharp_j\, e'$*
- *$(j,e) \prec \underline{(j',e')}$ iff $j = j'$ and $e \prec_j \underline{e'}$*
- *$(j,e) \rhd \underline{(j',e')}$ iff $e' \in F_{j'}$ and $j \neq j'$, or $e \rhd_j \underline{e'}$*

Example 3.3 (RPES coproduct)**.** *Given RPESs $\mathscr{E}_0 = (E_0, F_0, <_0, \sharp_0, \prec_0, \rhd_0)$ and $\mathscr{E}_1 = (E_1, F_1, <_1, \sharp_1, \prec_1, \rhd_1)$ where $E_0 = \{a,b\}$, $F_0 = \{a,b\}$, $a <_0 b$, $a \prec_0 \underline{b}$ and $E_1 = \{c,d\}$, $F_1 = \{c\}$, and $d \rhd_1 \underline{c}$, the coproduct $\mathscr{E}_0 + \mathscr{E}_1$ is $(E, F, <, \sharp, \prec, \rhd)$, where $E = \{(0,a),(0,b),(1,c),(1,d)\}$, $F = \{(0,a),(0,b),(1,c)\}$, $(0,a) < (0,b)$, $(0,a) \prec \underline{(0,b)}$, $(0,a) \,\sharp\, (1,c)$, $(0,a) \,\sharp\, (1,d)$, $(0,b) \,\sharp\, (0,c)$, $(0,b) \,\sharp\, (0,d)$, $(0,a) \rhd (1,c)$, $(0,b) \rhd (1,c)$, $(1,c) \rhd \overline{(0,a)}$, $(1,d) \rhd (0,a)$, $(1,c) \rhd (0,b)$, $(1,d) \rhd (0,b)$, and $(1,d) \rhd \underline{(1,c)}$.*

As we did with PESs, we will now add reversibility to AESs. *Reversible asymmetric event structures* (RAES) [13] (Definition 3.4) consist of events, some of which may be reversible, as well as causation and precedence, similar to an AES, except that $\prec$ is no longer a partial order, and instead just well-founded. In addition, both work on the reversible events, similarly to the RPES.

Definition 3.4 (RAES [13])**.** *A reversible asymmetric event structure (RAES) is a quadruple $\mathcal{E} = (E, F, \prec, \lhd)$ where E is the set of events, $F \subseteq E$ is the set of reversible events, and*

- *$\lhd \subseteq (E \cup \underline{F}) \times E$ is the irreflexive precedence relation*
- *$\prec \subseteq E \times (E \cup \underline{F})$ is the causation relation, which is irreflexive and well-founded, such that for all $\alpha \in E \cup \underline{F}$, $\{e \in E \mid e \prec \alpha\}$ is finite and has no $\lhd$-cycles, and for all $e \in F$, $e \prec \underline{e}$*
- *for all $e \in E$ and $\alpha \in E \cup \underline{F}$ if $e \prec \alpha$ then not $e \rhd \alpha$*
- *$e \prec\!\!\prec e'$ implies $e \lhd e'$, where $e \prec\!\!\prec e'$ means that $e \prec e'$ and if $e \in F$ then $e' \rhd \underline{e}$*
- *$\prec\!\!\prec$ is transitive and if $e \sharp e'$ and $e \prec\!\!\prec e''$ then $e'' \sharp e'$*

Once again we create a category **RAES** by defining RAES-morphisms. This definition is nearly identical to that of an AES-morphism, with the added condition that, like in the RPES morphism, reversible events can only synchronise with other reversible events. The category **RAES** has coproducts, defined very similarly to the RPES coproduct, though without symmetric conflict and combining both causation relations into one.

4 Reversible General Event Structures

The last kind of event structure we add reversibility to is the general event structure. The *reversible (general) event structure* differs from the general event structure, not only by allowing the reversal of events, but also by including a preventing set in the enabling relation, so that $X \obslash Y \vdash e$ means e is enabled in configurations that include all the events of X but none of the events of Y. An example of an RES can be seen in Figure 4b. In all examples we will use $X \obslash Y \vdash e^*$ as shorthand for $X' \obslash Y \vdash e^*$ whenever $X \subseteq X' \in \mathsf{Con}$

Definition 4.1 (RES [15])**.** *A reversible event structure (RES) is a triple $\mathcal{E} = (E, Con, \vdash)$ where E is the set of events, $Con \subseteq_{\text{fin}} 2^E$ is the consistency relation, which is left-closed, $\vdash \subseteq Con \times 2^E \times (E \cup \underline{E})$ is the enabling relation, and (1) if $X \obslash Y \vdash e^*$ then $(X \cup \{e\}) \cap Y = \emptyset$, (2) if $X \obslash Y \vdash \underline{e}$ then $e \in X$, and (3) if $X \obslash Y \vdash e^*$, $X \subseteq X' \in Con$, and $X' \cap Y = \emptyset$ then $X' \obslash Y \vdash e^*$.*

To define the category **RES**, we need to define a RES-morphism (Definition 4.2). With the exception of the requirements regarding preventing sets, it is identical to the definition of an ES-morphism. We treat the preventing set similarly to (asymmetric) conflict in PES, AES, RPES, and RAES-morphisms.

Definition 4.2 (RES morphism)**.** *Let $\mathcal{E}_0 = (E_0, Con_0, \vdash_0)$ and $\mathcal{E}_1 = (E_1, Con_1, \vdash_1)$ be RESs. A morphism $f : \mathcal{E}_0 \to \mathcal{E}_1$ is a partial function $f : E_0 \to E_1$ such that*

- *for all $e \in E_0$, if $f(e) \neq \bot$ and $X \obslash Y \vdash_0 e^*$ then there exists a $Y_1 \subseteq E_1$ such that for all $e_0 \in E_0$, if $f(e_0) \in Y_1$ then $e_0 \in Y$ and $f(X) \obslash Y_1 \vdash_1 f(e)^*$*
- *for any $X_0 \in Con_0$, $f(X_0) \in Con_1$*
- *for all $e, e' \in E_0$, if $f(e) = f(e') \neq \bot$ and $e \neq e'$ then no $X \in Con_0$ exists such that $e, e' \in X$*

As with **RPES** and **RAES**, **RES** has coproducts (Definition 4.3).

Definition 4.3 (RES coproduct)**.** *Given RESs $\mathscr{E}_0 = (E_0, \mathit{Con}_0, \vdash_0)$ and $\mathscr{E}_1 = (E_0, \mathit{Con}_0, \vdash_0)$, their coproduct $\mathscr{E}_0 + \mathscr{E}_1$ is $(E, \mathit{Con}, \vdash)$ where:*

- $E = \{(0,e) \mid e \in E_0\} \cup \{(1,e) \mid e \in E_1\}$
- *injections i_j exist such that for $e \in E_j$ $i_j(e) = (j,e)$ for $j \in \{0,1\}$*
- *$X \in \mathit{Con}$ iff $\exists X_0 \in \mathit{Con}_0 . i_0(X_0) = X$ or $\exists X_1 \in \mathit{Con}_1 . i_1(X_1) = X$*
- *$X \oslash Y \vdash (j,e)^*$ iff $\exists X_j, Y_j \in E_j$ such that $X_j \oslash Y_j \vdash e^*$, $i_j(X_j) = X$, $Y = i_j(Y_j) \cup (E \setminus i_j(E_j))$*

We also define the product of RESs (Definition 4.4). A product can be described as a parallel composition of two RESs. The reason we did not define the products of RPESs or RAESs is, that while the ES product defined in [19] easily translates to RESs, definitions of PES products, such as the one based on mapping the PESs into a domain and back seen in [17], are far more complex and difficult to translate directly to a reversible setting. Since we do not have mappings from CSs to RPESs or RAESs, this is not a possible solution. Example 4.5 shows the product of two RESs.

Definition 4.4 (RES product)**.** *Given RESs $\mathscr{E}_0 = (E_0, \mathit{Con}_0, \vdash_0)$ and $\mathscr{E}_1 = (E_1, \mathit{Con}_1, \vdash_1)$, their partially synchronous product $\mathscr{E}_0 \times \mathscr{E}_1$ is $(E, \mathit{Con}, \vdash)$ where:*

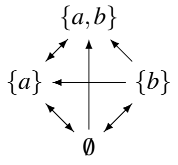
- $E = E_0 \times_* E_1 = \{(e,*) \mid e \in E_0\} \cup \{(*,e) \mid e \in E_1\} \cup \{(e,e') \mid e \in E_0 \text{ and } e' \in E_1\}$
- *there exist projections π_0, π_1 such that for $(e_0,e_1) \in E$, $\pi_i((e_0,e_1)) = e_i$*
- *$X \in \mathit{Con}$ if $\pi_0(X) \in \mathit{Con}_0$, $\pi_1(X) \in \mathit{Con}_1$, and for all $e,e' \in X$, if $\pi_0(e) = \pi_0(e')$ or $\pi_1(e) = \pi_1(e')$ then $e = e'$*
- *$X \oslash Y \vdash e^*$ if*
    - *if $\pi_0(e) \neq *$ then $\pi_0(X) \oslash \pi_0(Y) \vdash \pi_0(e)^*$*
    - *if $\pi_1(e) \neq *$ then $\pi_1(X) \oslash \pi_1(Y) \vdash \pi_1(e)^*$*
    - *if $e^* = \underline{e}$ then $e \in X$*

Example 4.5 (RES product)**.** *Given RESs $\mathscr{E}_0 = (E_0, \mathit{Con}_0, \vdash_0)$ and $\mathscr{E}_1 = (E_1, \mathit{Con}_1, \vdash_1)$, where $E_0 = \{a,b\}$, $\mathit{Con}_0 = 2^{E_0}$, $\emptyset \oslash \emptyset \vdash_0 a$, $\{a\} \oslash \emptyset \vdash_0 b$, $\{a,b\} \oslash \emptyset \vdash_0 \underline{b}$, and $\{a\} \oslash \emptyset \vdash_0 \underline{a}$ and $E_1 = \{c\}$, $\mathit{Con}_1 = \{\emptyset, \{c\}\}$, $\emptyset \oslash \emptyset \vdash_1 c$, and $\{c\} \oslash \vdash_1 \underline{c}$, the product $\mathscr{E}_0 \times \mathscr{E}_1$ is $(E, \mathit{Con}, \vdash)$ where $E = \{(a,*), (b,*), (a,c), (b,c), (*,c)\}$, $\mathit{Con} = \{\emptyset, \{(a,*)\}, \{(b,*)\}, \{(a,c)\}, \{(b,c)\}, \{(*,c)\}, \{(a,*),(b,*)\}, \{(a,*),(b,c)\}, \{(a,*),(*,c)\}, \{(a,c),(b,*)\}, \{(b,*),(*,c)\}, \{(a,*),(b,*),(*,c)\}\}$, $\emptyset \oslash \emptyset \vdash (a,*)$, $\{(a,*) \oslash \emptyset \vdash (b,*)$, $\{(a,c)\} \oslash \emptyset \vdash (b,*)$, $\emptyset \oslash \emptyset \vdash (a,c)$, $\{(a,*) \oslash \emptyset \vdash (b,c)$, $\{(a,c)\} \oslash \emptyset \vdash (b,c)$, $\emptyset \oslash \emptyset \vdash (*,c)$, $\{(a,*)\} \oslash \emptyset \vdash \underline{(a,*)}$, $\{(b,*),(a,*)\} \oslash \emptyset \vdash \underline{(b,*)}$, $\{(b,*),(a,c)\} \oslash \emptyset \vdash \underline{(b,*)}$, $\{(a,c)\} \oslash \emptyset \vdash \underline{(a,c)}$, $\{(b,c),(a,*)\} \oslash \emptyset \vdash \underline{(b,c)}$, and $\{(*,c)\} \oslash \emptyset \vdash \underline{(*,c)}$.*

We also create functors from **RPES** and **RAES** to **RES**. While not all AESs have ESs which map to the same domain, RAESs map into RESs using the preventing set to model asymmetric conflict as described in Definition 4.6.

Definition 4.6 (From **RAES** to **RES**)**.** *The mapping $P_{ar} : \textbf{RAES} \to \textbf{RES}$ is defined as:*

- *$P_{ar}(\mathscr{E}) = (E, \mathit{Con}, \vdash)$ where*
  $\mathit{Con} = \{X \subseteq E \mid \lhd \text{ is well-founded on } X\}$
  $X \oslash Y \vdash e$ if $\{e' \mid e' \prec e\} \subseteq X \in \mathit{Con}$, $Y = \{e' \mid e' \rhd e\}$, $X \cap Y = \emptyset$, and $e \in E$
  $X \oslash Y \vdash \underline{e}$ if $\{e' \mid e' \prec \underline{e}\} \subseteq X \in \mathit{Con}$, $Y = \{e' \mid e' \rhd \underline{e}\}$, $X \cap Y = \emptyset$, and $e \in F$
- *$P_{ar}(f) = f$*



(a) CS $\mathscr{C}$

$\mathscr{E} = (E, \mathsf{Con}, \vdash)$ where
$E = \{a,b\}$
$\mathsf{Con} = \{\emptyset, \{a\}, \{b\}, \{a,b\}\}$
$\emptyset \oslash \emptyset \vdash a$, $\emptyset \oslash \emptyset \vdash b$,
$\{b\} \oslash \emptyset \vdash \underline{b}$, $\{a\} \oslash \{b\} \vdash \underline{a}$

(b) RES $\mathscr{E}$

Figure 4: A CS and the corresponding RES such that $R(\mathscr{C}) = \mathscr{E}$ and $C_r(\mathscr{E}) = \mathscr{C}$.

5 Configuration Systems

Configuration systems perform a similar role in the reversible setting to domains in the forward-only setting, though they have a more operational character. A configuration system [13] (Definition 5.1) consists of a set of events, E, some of which, F, are reversible, a set C of configurations on these, and an optionally labelled transition relation $\rightarrow$ such that if $X \xrightarrow{A\cup\underline{B}} Y$ then the events of A can happen and the events of B can be undone in any order starting from configuration X, resulting in Y. We also leave out Y when describing such a transition, since it is implied that $Y = (X \setminus B) \cup A$. A CS is shown in Figure 4a.

Definition 5.1 (Configuration system [13]). *A configuration system (CS) is a quadruple $\mathscr{C} = (E, F, \mathsf{C}, \rightarrow)$ where E is a set of events, $F \subseteq E$ is a set of reversible events, $\mathsf{C} \subseteq 2^E$ is the set of configurations, and $\rightarrow \subseteq \mathsf{C} \times 2^{E\cup\underline{F}} \times \mathsf{C}$ is an optionally labelled transition relation such that if $X \xrightarrow{A\cup\underline{B}} Y$ then:*

- *$A \cap X = \emptyset$, $B \subseteq X \cap F$, and $Y = (X \setminus B) \cup A$*
- *for all $A' \subseteq A$ and $B' \subseteq B$, we have $X \xrightarrow{A'\cup\underline{B'}} Z \xrightarrow{(A\setminus A')\cup\underline{(B\setminus B')}} Y$, meaning $Z = (X \setminus B') \cup A' \in \mathsf{C}$*

We define a notion of morphism (Definition 5.2), creating the category **CS**.

Definition 5.2 (CS-morphism). *Let $\mathscr{C}_0 = (E_0, F_0, \mathsf{C}_0, \rightarrow_0)$ and $\mathscr{C}_1 = (E_1, F_1, \mathsf{C}_1, \rightarrow_1)$ be configuration systems. A configuration system morphism is a partial function $f : E_0 \rightarrow E_1$ such that*

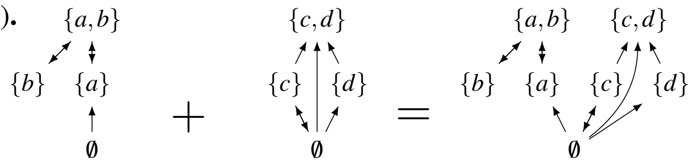
- *for any $X, Y \in \mathsf{C}_0$, $A \subseteq E_0$, and $B \subseteq F_0$, if $X \xrightarrow{A\cup\underline{B}}_0 Y$ then $f(X) \xrightarrow{f(A)\cup f(\underline{B})}_1 f(Y)$*
- *for any $X \in \mathsf{C}_0$, $f(X) \in \mathsf{C}_1$*
- *for all $e_0, e_0' \in E_0$, if $f(e_0) = f(e_0') \neq \bot$ and $e_0 \neq e_0'$ then there exists no $X \in \mathsf{C}_0$ such that $e_0, e_0' \in X$*

We also define the coproduct of two CSs (Definition 5.3). This is illustrated with CSs modelling the RPESs and RESs from Examples 3.3 and 4.5 in Example 5.4.

Definition 5.3 (CS coproduct). *Given CSs $\mathscr{C}_0 = (E_0, F_0, \mathsf{C}_0, \rightarrow_0)$ and $\mathscr{C}_1 = (E_1, F_1, \mathsf{C}_1, \rightarrow_1)$, their coproduct $\mathscr{C}_0 + \mathscr{C}_1 = (E, F, \mathsf{C}, \rightarrow)$ where:*

- *$E = \{(0,e) \mid e \in E_0\} \cup \{(1,e) \mid e \in E_1\}$ and $F = \{(0,e) \mid e \in F_0\} \cup \{(1,e) \mid e \in F_1\}$*
- *injections i_j exist such that for $e \in E_j$ $i_j(e) = (j,e)$ for $j \in \{0,1\}$*
- *$X \in \mathsf{C}$ iff $\exists X_0 \in \mathsf{C}_0 . i_0(X_0) = X$ or $\exists X_1 \in \mathsf{C}_1 . i_1(X_1) = X$*
- *$X \xrightarrow{A\cup\underline{B}} Y$ iff there exists $j \in \{0,1\}$ such that there exist $X_j, Y_j, A_j, B_j \subseteq E_j$ such that $i_j(X_j) = X$, $i_j(Y_j) = Y$, $i_j(A_j) = A$, $i_j(B_j) = B$, and $X_j \xrightarrow{A_j\cup\underline{B_j}}_j Y_j$.*

Example 5.4 (Coproduct).

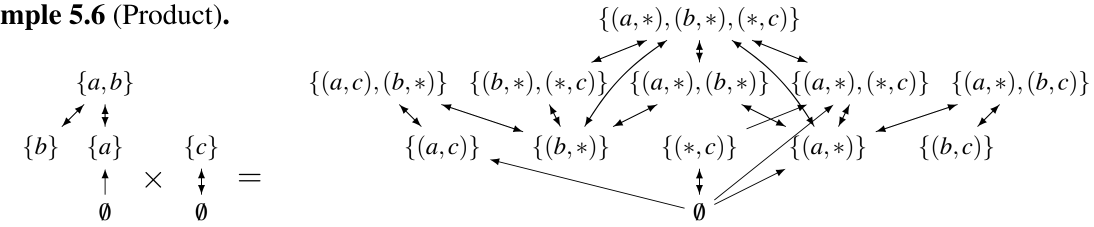


We also define the product of CSs (Definition 5.5). This is illustrated in Example 5.6, where the CSs represent the RESs of Example 4.5.

Definition 5.5 (CS product). *Given CSs $\mathscr{C}_0 = (E_0, F_0, C_0, \rightarrow_0)$ and $\mathscr{C}_1 = (E_1, F_1, C_1, \rightarrow_1)$, their partially synchronous product $\mathscr{C}_0 \times \mathscr{C}_1 = (E, F, C, \rightarrow)$ where:*

- $E = E_0 \times_* E_1 = \{(e,*) \mid e \in E_0\} \cup \{(*,e) \mid e \in E_1\} \cup \{(e,e') \mid e \in E_0 \text{ and } e' \in E_1\}$
- $F = F_0 \times_* F_1 = \{(e,*) \mid e \in F_0\} \cup \{(*,e) \mid e \in F_1\} \cup \{(e,e') \mid e \in F_0 \text{ and } e' \in F_1\}$
- *there exist projections π_0, π_1 such that for $(e_0, e_1) \in E$, $\pi_i((e_0, e_1)) = e_i$*
- *$X \in C$ if $\pi_0(X) \in C_0$, $\pi_1(X) \in C_1$, and for all $e, e' \in X$, if $\pi_0(e) = \pi_0(e')$ or $\pi_1(e) = \pi_1(e')$ then $e = e'$*
- *$X \xrightarrow{A \cup \underline{B}} Y$ if $B \subseteq X$ and*
  - *if $\pi_0(A \cup B) \neq \emptyset$ then $\pi_0(X) \xrightarrow{\pi_0(A) \cup \underline{\pi_0(B)}}_0 \pi_0(Y)$*
  - *if $\pi_1(A \cup B) \neq \emptyset$ then $\pi_1(X) \xrightarrow{\pi_1(A) \cup \underline{\pi_1(B)}}_1 \pi_1(Y)$*

Example 5.6 (Product).



We define a functor C_r from **RES** to **CS** (Definition 5.7).

Definition 5.7 (From **RES** to **CS**). *The mapping $C_r : \boldsymbol{RES} \rightarrow \boldsymbol{CS}$ is defined as*

- *$C_r((E, Con, \vdash)) = (E, F, C, \rightarrow)$, where (1) $e \in F$ if there exists X, Y such that $X \oslash Y \vdash \underline{e}$, (2) $C \in C$ if for all $X \subseteq_{\text{fin}} C$, $X \in Con$, and (3) for $X, Y \in C$, $X \xrightarrow{A \cup \underline{B}} Y$ if*
  - *$Y = (X \setminus B) \cup A$, $A \cap X = \emptyset$, $B \subseteq X$, and $X \cup A \in C$*
  - *for all e in A, $X' \oslash Z \vdash e$ for some X', Z such that $X' \subseteq_{\text{fin}} X \setminus B$ and $Z \cap (X \cup A) = \emptyset$*
  - *for all $e \in B$, $X' \oslash Z \vdash \underline{e}$ for some X', Z such that $X' \subseteq_{\text{fin}} X \setminus (B \setminus \{e\})$ and $Z \cap (X \cup A) = \emptyset$*
- *$C_r(f) = f$*

Applying this functor to a RES results in a *finitely enabled CS* (FCS), that is to say a CS such that there does not exist a transition from an infinite configuration $X \xrightarrow{A \cup \underline{B}}$, such that there does not exist a finite configuration $X' \subseteq_{\text{fin}} X$ such that $X' \xrightarrow{A \cup \underline{B}}$ and whenever $X' \subseteq X'' \subseteq X$ there exists a transition $X'' \xrightarrow{A \cup \underline{B}}$. We call the category of these CSs and the CS-morphisms between them **FCS**, and describe a functor, R, from this category to **RES** in Definition 5.8. An example of C_r and R can be seen in Figure 4.

Definition 5.8 (From **FCS** to **RES**). *The mapping $R : \boldsymbol{FCS} \rightarrow \boldsymbol{RES}$ is defined as:*

- *$R((E, F, C, \rightarrow)) = (E, Con, \vdash)$ where $X \in Con$ if $X \subseteq_{\text{fin}} C \in C$ and:*

  - *If* $X \xrightarrow{\{e^*\}}$ *and*
    - $X' \subseteq X$, $X' \xrightarrow{\{e^*\}}$, *and whenever* $X' \subseteq X'' \subseteq X$ *there exists a transition* $X'' \xrightarrow{\{e^*\}}$
    - *no* $X'' \subset X'$ *exists such that* $X'' \xrightarrow{\{e^*\}}$, *and whenever* $X'' \subseteq X''' \subseteq X$ *there exists a transition* $X''' \xrightarrow{\{e^*\}}$
    - *no* $X'' \supset X$ *exists such that* $X'' \xrightarrow{\{e^*\}}$, *and whenever* $X' \subseteq X''' \subseteq X''$ *there exists a transition* $X''' \xrightarrow{\{e^*\}}$

    then
    - *if* $e^* = e$, *then for all* $X'' \in \mathsf{Con}$ *such that* $X' \subseteq X'' \subseteq X \cup \{e\}$, $X'' \oslash E \setminus X \cup \{e\} \vdash e$
    - *if* $e^* = \underline{e}$, *then for all* $X'' \in \mathsf{Con}$ *such that* $X' \subseteq X'' \subseteq X$, $X'' \oslash E \setminus (X \setminus \{e\} \vdash \underline{e}$

- $R(f) = f$

As Theorem 5.9 states, C_r and R are in many cases inverses of each other. If $\mathscr{C} = (E, F, \mathsf{C}, \rightarrow)$ is a CS, then $C_r(R(\mathscr{C})) = \mathscr{C}' = (E, F', \mathsf{C}', \rightarrow)$ where C' is the downwards closure of C and $F' \subseteq F$ is the set of events such that a transition $X \xrightarrow{\{e\}}$ exists.

Theorem 5.9. *Given an FCS* $\mathscr{C} = (E, F, C, \rightarrow)$, $C_r(R(\mathscr{C})) = \mathscr{C}$ *if* C *is downwards closed, and for all* $e \in F$ *there exists a transition* $X \xrightarrow{e}$. *If* $\mathscr{E} = (E, \mathsf{Con}, \vdash)$ *is an RES with no "unnecessary" enablings* $X \oslash Y' \vdash e^*$ *such that* $X \oslash Y \vdash e^*$ *for* $Y \subset Y'$ *then* $R(C_r(\mathscr{E})) = \mathscr{E}$.

6 Stable Reversible Event Structures and Configuration Systems

Similarly to the stable event structures, we define the *stable reversible event structures* (Definition 6.1), and create the category **SRES** consisting of SRESs and the RES-morphisms between them. SRESs and SESs are defined identically, with the exception that in an SRES the preventing sets are included as well, and treated in much the same way as the enabling sets. Like in a SES, an event in a configuration of a SRES will have one possible cause as long as the configuration has been reached by only going forwards.

Definition 6.1 (Stable RES)**.** *A stable reversible event structure (SRES) is an RES* $\mathscr{E} = (E, \mathsf{Con}, \vdash)$ *such that for all* $e^* \in E$ *if* $X \oslash Y \vdash e^*$, $X' \oslash Y' \vdash e^*$, *and* $X \cup X' + e^* \in \mathsf{Con}$ *then* $X \cap X' \oslash Y \cap Y' \vdash e^*$.

Similarly, we can define a *stable configuration system* (Definition 6.2). This has the property that if $\mathscr{E}$ is a SRES then $C_r(\mathscr{E})$ is a SCS, and if $\mathscr{C}$ is a SCS then $R(\mathscr{C})$ is a SRES.

Definition 6.2 (Stable CS)**.** *A stable CS (SCS) is an FCS* $\mathscr{C} = (E, F, C, \rightarrow)$ *such that*

1. C *is downwards closed*
2. *For all* $e \in F$, *there exists a transition* $X \xrightarrow{e}$
3. *For* $X_1, X_2, X_3 \in C$:
   (a) *if* $X_1 \subseteq X_2 \subseteq X_3$, $X_1 \xrightarrow{A \cup \underline{B}}$, *and* $X_3 \xrightarrow{A \cup \underline{B}}$, *then* $X_2 \xrightarrow{A \cup \underline{B}}$
   (b) *if* $((X_1 \cup X_2) \setminus B) \cup A \in C$, $X_1 \xrightarrow{A \cup \underline{B}}$, *and* $X_2 \xrightarrow{A \cup \underline{B}}$, *then* $X_1 \cup X_2 \xrightarrow{A \cup \underline{B}}$ *and* $X_1 \cap X_2 \xrightarrow{A \cup \underline{B}}$
   (c) *if* $X_0, X_1, X_2, X_3 \in C$, $A_0, A_1, B_0, B_1 \subseteq E$ *and there exist transitions* $X_0 \xrightarrow{A_0 \cup \underline{B_0}} X_1$, $X_0 \xrightarrow{A_1 \cup \underline{B_1}} X_2$, $X_1 \xrightarrow{A_1 \cup \underline{B_1}} X_3$, *and* $X_2 \xrightarrow{A_0 \cup \underline{B_0}} X_3$, *then* $X_0 \xrightarrow{A_0 \cup A_1 \cup \underline{B_0 \cup B_1}} X_3$

Figure 4a shows a stable CS. One way to make it not stable would be to remove the transition from $\emptyset$ to $\{a, b\}$, since that would violate Item 3c.

As [13] did for RPESs and RAESs, we define a subcategory of *cause-respecting* RESs in Definition 6.4. This is based on the idea that if e' enables e, then e' cannot be reversed from a configuration which does not have another possible enabling set for e. Unlike causal reversibility [5] however, a configuration fulfilling these conditions does not guarantee that reversing is possible.

Definition 6.3 (Minimal enabling configurations for RES $m_{RES}(e)$)**.** *Given an RES $\mathcal{E} = (E, \mathsf{Con}, \vdash)$ the set of minimal enabling configurations of an event $e \in E$ is defined as:*
$m_{RES}(e) = \{X \in \mathsf{Con} \mid \exists Y. X \oslash Y \vdash e \text{ and } \forall X', Y'. X' \oslash Y' \vdash e \Rightarrow X' \not\subset X\}$

Definition 6.4 (CRES)**.** *A CRES $\mathcal{E} = (E, \mathsf{Con}, \vdash)$ is an RES such that for all $e, e' \in E$, $e' \in X \in m_{RES}(e)$ iff whenever $X' \oslash Y' \vdash \underline{e'}$, we have $e \in Y'$ or there exists an $X'' \subseteq X' \setminus \{e'\}$ such that $X'' \in m_{RES}(e)$.*

Moreover we define a *cause-respecting* CS in much that same way as a CRES (Definition 6.6). This has the property that if $\mathcal{E}$ is a CRES then $C_r(\mathcal{E})$ is a CCS, and if $\mathcal{C}$ is a finitely enabled CCS then $R(\mathcal{C})$ is a CRES. We can then prove Theorem 6.8, which is analogous to a property of cause-respecting RPESs and RAESs proved in [13]. The CS in Figure 4a is cause-respecting, but removing the transition from $\emptyset$ to $\{a\}$ would change that.

Definition 6.5 (Minimal enabling configurations for CS $m_{CS}(e)$)**.** *Given a CS $\mathcal{C} = (E, F, C, \rightarrow)$ the set of minimal enabling configurations of an event $e \in E$ is defined as*

$$m_{CS}(e) = \{X \in \mathsf{C} \mid X \xrightarrow{\{e\}} \text{ and } \forall X'. X' \xrightarrow{\{e\}} \Rightarrow X' \not\subset X\}$$

Definition 6.6 (CCS)**.** *A cause-respecting CS $\mathcal{C} = (E, F, C, \rightarrow)$ is a CS such that if $e' \in X \in m_{CS}(e)$, then whenever $X' \xrightarrow{\{\underline{e'}\}} Y'$ and $e \in X'$, there exists an $X'' \subseteq Y'$ such that $X'' \in m_{CS}(e)$.*

Proposition 6.7. *If $\mathcal{E}$ is a CSRES then $C_r(\mathcal{E})$ is a CSCS, and if $\mathcal{C}$ is a CSCS then $R(\mathcal{C})$ is a CSRES.*

Theorem 6.8. *If $\mathcal{C} = (E, F, C, \rightarrow)$ is a CSCS then every reachable configuration is forwards reachable.*

7 Conclusion

We have defined categories for configuration systems (CS), reversible prime event structures (RPES), reversible asymmetric event structures (RAES), and reversible general event structures (RES), and functors between them, showing all the event structures can be modelled as CSs and finitely enabled CSs can be modelled as RESs in way that preserves morphisms (Theorem 5.9). We also defined coproducts for each of these categories, though products only for RESs and CSs.

With a view to the semantics of causal reversible process calculi, we have also defined stable and cause-respecting subcategories of RESs, in which every reachable configuration is forwards reachable (Theorem 6.8).

Future Work: Defining a product of RPESs and RAESs will likely be trickier than for RESs, since definitions of products of prime event structures are far more complex than those of general event structures [17], and we note that the product of asymmetric event structures is as yet undefined. We plan to formulate a notion of 'causal' RES which strengthens the 'cause-respecting' safety condition with a liveness condition.

Acknowledgements: We thank the referees for their helpful comments. This work was partially supported by EPSRC DTP award; EPSRC projects EP/K034413/1, EP/K011715/1, EP/L00058X/1, EP/N027833/1 and EP/N028201/1; EU FP7 612985 (UPSCALE); and EU COST Action IC1405.

References

[1] Paolo Baldan, Andrea Corradini & Ugo Montanari (2001): *Contextual Petri Nets, Asymmetric Event Structures, and Processes*. Information and Computation 171(1), pp. 1 – 49, doi:http://dx.doi.org/10.1006/inco.2001.3060. Available at `http://www.sciencedirect.com/science/article/pii/S0890540101930603`.

[2] William J Bowman, Roshan P James & Amr Sabry (2011): *Dagger traced symmetric monoidal categories and reversible programming*. In: Workshop on Reversible Computation, RC 2011. Available at `https://williamjbowman.com/resources/cat-rev.pdf`.

[3] Ioana Cristescu, Jean Krivine & Daniele Varacca (2013): *A Compositional Semantics for the Reversible pi-Calculus*. In: IEEE Symposium on Logic in Computer Science, LICS '13, IEEE Computer Society, Washington, DC, USA, pp. 388–397, doi:10.1109/LICS.2013.45. Available at `http://dx.doi.org/10.1109/LICS.2013.45`.

[4] Ioana Cristescu, Jean Krivine & Daniele Varacca (2016): *Rigid Families for the Reversible π-Calculus*. In: Reversible Computation - 8th International Conference, RC 2016, Bologna, Italy, July 7-8, 2016, Proceedings, Lecture Notes in Computer Science 9720, Springer, pp. 3–19, doi:10.1007/978-3-319-40578-0_1. Available at `http://dx.doi.org/10.1007/978-3-319-40578-0_1`.

[5] Vincent Danos & Jean Krivine (2004): *Reversible Communicating Systems*. In Philippa Gardner & Nobuko Yoshida, editors: CONCUR, LNCS 3170, Springer, Berlin, Heidelberg, pp. 292–307, doi:10.1007/978-3-540-28644-8_19. Available at `http://dx.doi.org/10.1007/978-3-540-28644-8_19`.

[6] Vincent Danos & Jean Krivine (2007): *Formal Molecular Biology Done in CCS-R*. Electronic Notes in Theoretical Computer Science 180(3), pp. 31 – 49, doi:http://dx.doi.org/10.1016/j.entcs.2004.01.040. Available at `//www.sciencedirect.com/science/article/pii/S1571066107004963`. BioConcur.

[7] Vincent Danos, Jean Krivine & Paweł Sobociński (2007): *General Reversibility*. In: EXPRESS, 175(3), pp. 75 – 86, doi:http://dx.doi.org/10.1016/j.entcs.2006.07.036. Available at `http://www.sciencedirect.com/science/article/pii/S1571066107003775`.

[8] Ivan Lanese, Claudio Antares Mezzina & Jean-Bernard Stefani (2010): *Reversing Higher-Order Pi*. In Paul Gastin & François Laroussinie, editors: CONCUR, LNCS 6269, Springer, Berlin, Heidelberg, pp. 478–493, doi:10.1007/978-3-642-15375-4_33. Available at `http://dx.doi.org/10.1007/978-3-642-15375-4_33`.

[9] Ivan Lanese, Claudio Antares Mezzina & Francesco Tiezzi (2014): *Causal-Consistent Reversibility*. Bulletin of the EATCS 114, p. 17. Available at `https://hal.inria.fr/hal-01089350`.

[10] Mogens Nielsen, Gordon Plotkin & Glynn Winskel (1979): *Petri nets, event structures and domains*. In Gilles Kahn, editor: Semantics of Concurrent Computation, LNCS 70, Springer, Berlin, Heidelberg, pp. 266–284, doi:10.1007/BFb0022474. Available at `http://dx.doi.org/10.1007/BFb0022474`.

[11] Iain Phillips & Irek Ulidowski (2006): *Reversing Algebraic Process Calculi*. In Luca Aceto & Anna Ingólfsdóttir, editors: FOSSACS, LNCS 3921, Springer, Berlin, Heidelberg, pp. 246–260, doi:10.1007/11690634_17. Available at `http://dx.doi.org/10.1007/11690634_17`.

[12] Iain Phillips & Irek Ulidowski (2007): *Reversibility and Models for Concurrency*. Electr. Notes Theor. Comput. Sci. 192(1), pp. 93–108, doi:10.1016/j.entcs.2007.08.018. Available at `http://dx.doi.org/10.1016/j.entcs.2007.08.018`.

[13] Iain Phillips & Irek Ulidowski (2015): *Reversibility and asymmetric conflict in event structures*. Journal of Logical and Algebraic Methods in Programming 84(6), pp. 781 – 805, doi:http://dx.doi.org/10.1016/j.jlamp.2015.07.004. Available at `http://www.sciencedirect.com/science/article/pii/S2352220815000619`. Special Issue on Open Problems in Concurrency Theory.

[14] Iain Phillips, Irek Ulidowski & Shoji Yuen (2013): *A Reversible Process Calculus and the Modelling of the ERK Signalling Pathway*. In Robert Glück & Tetsuo Yokoyama, editors: RC, LNCS 7581, Springer,

Berlin, Heidelberg, pp. 218–232, doi:10.1007/978-3-642-36315-3_18. Available at http://dx.doi.org/10.1007/978-3-642-36315-3_18.

[15] Iain Phillips, Irek Ulidowski & Shoji Yuen (2013): *Modelling of Bonding with Processes and Events*. In Gerhard W. Dueck & D. Michael Miller, editors: *RC*, *LNCS* 7948, Springer, Berlin, Heidelberg, pp. 141–154, doi:10.1007/978-3-642-38986-3_12. Available at http://dx.doi.org/10.1007/978-3-642-38986-3_12.

[16] Irek Ulidowski, Iain Phillips & Shoji Yuen (2014): *Concurrency and Reversibility*. In Shigeru Yamashita & Shin-ichi Minato, editors: *RC*, *LNCS* 8507, Springer, Cham, pp. 1–14, doi:10.1007/978-3-319-08494-7_1. Available at http://dx.doi.org/10.1007/978-3-319-08494-7_1.

[17] Frits W Vaandrager (1989): *A simple definition for parallel composition of prime event structures*. CS R 8903, Centre for Mathematics and Computer Science, P.O. Box 4079, 1009 AB Amsterdam, The Netherlands.

[18] Glynn Winskel (1982): *Event structure semantics for CCS and related languages*. In Mogens Nielsen & Erik Meineche Schmidt, editors: *ICALP*, *LNCS* 140, Springer, Berlin, Heidelberg, pp. 561–576, doi:10.1007/BFb0012800. Available at http://dx.doi.org/10.1007/BFb0012800.

[19] Glynn Winskel (1987): *Event structures*. In W. Brauer, W. Reisig & G. Rozenberg, editors: *Petri Nets: Applications and Relationships to Other Models of Concurrency. ACPN*, *LNCS* 255, Springer, Berlin, Heidelberg, pp. 325–392, doi:10.1007/3-540-17906-2_31. Available at http://dx.doi.org/10.1007/3-540-17906-2_31.