

Event Structures with Symmetry

Glynn Winskel ¹

University of Cambridge Computer Laboratory, England

Abstract

A category of event structures with symmetry is introduced and its categorical properties investigated. Applications to the event-structure semantics of higher order processes, nondeterministic dataflow and the unfolding of Petri nets with multiple tokens are sketched.

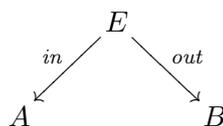
Keywords: Event structures, symmetry, pseudo monads, spans, higher order processes, nondeterministic dataflow, Petri net unfolding.

1 Introduction

In the paper introducing event structures [15] a ‘curious mismatch’ was noted. There event structures represent domains, so types. But they also represent processes which belong to a type. How are we to reconcile these two views?

One answer has arisen in recent work under the banner of ‘domain theory for concurrency’ (see [17] for a summary). This slogan stands for an attempt to push the methodology of domain theory and denotational semantics into the areas of interactive/concurrent/distributed computation, where presently more syntactic, operational or more informal methodologies prevail. Certain generalized relations (profunctors [4]) play a strong unifying role and it was discovered that in several contexts that they could be represented in a more informative operational way by spans of event structures [16,28,19].

A span of event structures is typically of the form



¹ Email: Glynn.Winskel@cl.cam.ac.uk

where *in* and *out* are maps of event structures—the maps are not necessarily of the same kind. The event structure E represents a process computing from an input type, represented by the event structure A , to output type represented by B . A span with no input amounts to just a single map $E \xrightarrow{out} B$ which we can read as expressing that the process E has type B . So spans are a way to reconcile the double role that event structures can take, as processes and as types.

Of course spans should compose. So one would like systematic ways to vary the *in* and *out* maps of spans which ensure they do. One way is to derive the maps by a Kleisli construction from monads on a fundamental category of event structures. With respect to suitable monads S and T satisfying a suitable distributivity law, one can form a bicategory of more general spans

$$\begin{array}{ccc} & E & \\ & \swarrow & \searrow \\ S(A) & & T(B) \end{array} .$$

It becomes important that event structures are able to support a reasonable repertoire of monads, including monads which produce multiple, essentially similar, copies of an event structure. For this the introduction of symmetry seems essential.²

In fact, there are several reasons for introducing symmetry to event structures and related models:

- It's there—at least informally. Symmetry often plays a role in the analysis of distributed algorithms. In particular, symmetry has always been present at least informally in the model of strand spaces, and has recently been exploited in exploring their behaviour [8], and was used to understand their expressivity [6]. Strand spaces are forms of event structures used in the analysis of security protocols. They comprise a collection of strands of input and output events, possibly with the generation of fresh values. Most often there are collections of strands which are essentially indistinguishable and can be permuted one for another without changing the strand space's behaviour.
- To obtain categorical characterizations of unfoldings of Petri nets in which places may hold with multiplicity greater than one. There are well-known ways to unfold such general nets; for example by distinguishing the tokens through 'colours,' splitting the places and events accordingly and reducing the problem to the unfolding in [15]. But the folding maps are not unique (w.r.t. an obvious cofreeness property). They are however unique 'up to symmetry.'
- Event structures are sometimes criticized for not being abstract enough. One precise way in which this manifests itself is that the category of event structures does not support monads and comonads of the kind discovered for more general presheaf models [4]. The computation paths of an event structure, its configurations, are ordered by inclusion. In contrast the paths of presheaf models can be

² Symmetry was introduced into game semantics specifically to support a 'copying' comonad [1].

related more generally by maps. Some (co)monads used for presheaf models allow the explicit copying of processes and produce a proper category of paths even when starting with a partial order of paths—this arises because of the similarity of one copy of a process with another.

The last point is especially pertinent to the versatility of spans of event structures. This paper presents a definition of a symmetry on an event structure. Roughly a symmetry will express the similarity of *finite* behaviours of an event structure. The introduction of symmetries to event structures will, in effect, put the structure of a category on their finite configurations, and so broaden the structure of computation paths event structures can represent. The ensuing category of event structures with symmetries will support a much richer class of (pseudo) monads, from which we can then obtain more general kinds of span. The category of event structures with symmetry with rigid maps emerges as fundamental; other maps on event structures can be obtained by a Kleisli construction or as instances of general spans starting from rigid maps.

Several applications, to be developed further in future work, are outlined in Section 8:

- *Event types*: One reason why so-called ‘interleaving’ models for concurrency have gained prevalence is that they support definitions by cases on the initial actions processes can do; another is that they readily support higher-order processes. Analogous facilities are lacking, at least in any reasonable generality, in ‘true-concurrency’ models—models like Petri nets and event structures, in which causal dependence and independence are represented explicitly. It is sketched how processes can be associated with ‘event types’ which specify the kinds of events they can do, and how event types can support definitions by cases on events. Much more needs to be done. But the examples do demonstrate the key role that symmetry and the copying of processes can play in obtaining flexible event types and event-based definitions.
- *Nondeterministic dataflow and affine-HOPLA*: ‘Stable’ spans of event structures, a direct generalisation of Berry’s stable functions [2], have been used to give semantics to nondeterministic dataflow [19] and the higher-order process language affine-HOPLA [16]. Stable spans can be obtained as instances of general spans. The realization of the ‘demand’ maps used there as a Kleisli construction on rigid maps provides a striking example of the power of symmetry.
- *Petri-net unfoldings*: One obvious application is to the unfolding of a general Petri net to an event structure with symmetry; the symmetry reflects that present in the original net through the interchangeability of tokens. Another related issue is the unfolding of higher-dimensional automata, where identifications of edges are reflected in the symmetry of the events to which they unfold.

2 Event structures

Event structures [15,22,25,26] are a model of computational processes. They represent a process, or system, as a set of event occurrences with relations to express how events causally depend on others, or exclude other events from occurring. In one of their simpler forms they consist of a set of events on which there is a consistency relation expressing when events can occur together in a history and a partial order of causal dependency—writing $e' \leq e$ if the occurrence of e depends on the previous occurrence of e' .

An *event structure* comprises (E, Con, \leq) , consisting of a countable³ set E , of *events* which are partially ordered by \leq , the *causal dependency relation*, and a *consistency relation* Con consisting of finite subsets of E , which satisfy

$$\begin{aligned} \{e' \mid e' \leq e\} \text{ is finite for all } e \in E, \\ \{e\} \in \text{Con for all } e \in E, \\ Y \subseteq X \in \text{Con} \Rightarrow Y \in \text{Con}, \text{ and} \\ X \in \text{Con} \ \& \ e \leq e' \in X \Rightarrow X \cup \{e\} \in \text{Con}. \end{aligned}$$

The events are to be thought of as event occurrences without significant duration; in any history an event is to appear at most once. We say that events e, e' are *concurrent* if $\{e, e'\} \in \text{Con}$ & $e \not\leq e'$ & $e' \not\leq e$. Concurrent events can occur together, independently of each other.

An event structure represents a process. A configuration is the set of all events which may have occurred by some stage, or history, in the evolution of the process. According to our understanding of the consistency relation and causal dependency relations a configuration should be consistent and such that if an event appears in a configuration then so do all the events on which it causally depends. Here we restrict attention to finite configurations.

The *configurations*, $\mathcal{C}(E)$, of an event structure E consist of those subsets $x \subseteq E$ which are

Consistent: $\forall X \subseteq x. X \text{ is finite} \Rightarrow X \in \text{Con}$, and

Down-closed: $\forall e, e'. e' \leq e \in x \Rightarrow e' \in x$.

We write $\mathcal{C}^o(E)$ for the *finite* configurations of the event structure E .

The configurations of an event structure are ordered by inclusion, where $x \subseteq x'$, *i.e.* x is a sub-configuration of x' , means that x is a sub-history of x' . Note that an individual configuration inherits an order of causal dependency on its events from the event structure so that the history of a process is captured through a partial order of events. The finite configurations correspond to those events which have occurred by some finite stage in the evolution of the process, and so describe the possible (finite) states of the process. The axioms on the consistency relation

³ In fact, that the events are countable can be replaced by the weaker condition of *consistent-countable*, which suffices for the proofs here and allows extra results, *e.g.* in Sections 8.2 and 8.4. An event structure is consistent-countable if there is a function χ from its events to the natural numbers ω such that $\{e_1, e_2\} \in \text{Con}$ and $\chi(e_1) = \chi(e_2)$ implies $e_1 = e_2$.

ensure that the down-closure of any finite set in the consistency relation is a finite configuration and that any event appears in a configuration: given $X \in \text{Con}$ its down-closure $\{e' \in E \mid \exists e \in X. e' \leq e\}$ is a finite configuration; in particular, for an event e , the set $[e] =_{def} \{e' \in E \mid e' \leq e\}$ is a configuration describing the whole causal history of the event e .

When the consistency relation is determined by the pairwise consistency of events we can replace it by a binary relation or, as is more usual, by a complementary binary conflict relation on events. It can be awkward to describe operations such as certain parallel compositions directly on the simple event structures here, essentially because an event determines its whole causal history. One closely related and more versatile, though perhaps less intuitive and familiar, model is that of stable families, described in Appendix C. Stable families will play an important technical role, both in establishing the existence of constructions such as products and pull-backs of event structures, and in providing more concrete ways to understand the introduction of symmetry to event structures and its consequences.

Let E and E' be event structures. A *partial map* of event structures $f : E \rightarrow E'$ is a partial function on events $f : E \rightarrow E'$ such that for all $x \in \mathcal{C}^o(E)$ its direct image $fx \in \mathcal{C}^o(E')$ and

$$\text{if } e_1, e_2 \in x \text{ and } f(e_1) = f(e_2) \text{ (with both defined), then } e_1 = e_2.$$

The partial map expresses how the occurrence of an event e in E induces the coincident occurrence of the event $f(e)$ in E' whenever it is defined. The partial function f respects the instantaneous nature of events: two distinct event occurrences which are consistent with each other cannot both coincide with the occurrence of a common event in the image. (The maps defined are unaffected if we allow all, not just finite, configurations, in the definition above.) Partial maps of event structures compose as partial functions.

For any event e a partial map of event structures $f : E \rightarrow E'$ must send the configuration $[e]$ to the configuration $f[e]$. Consequently the map f reflects causal dependency: whenever $f(e)$ and $f(e')$ are both defined with $f(e') \leq f(e)$, then $e' \leq e$. It follows that partial maps preserve the concurrency relation, when defined.

We will say the map is *total* iff the function f is total. Notice that for a total map f the condition on maps now says it is *locally injective*, in the sense that w.r.t. any configuration x of the domain the restriction of f to a function from x is injective; the restriction of f to a function from x to fx is thus bijective.

We say the map f is *rigid* iff it is total and for all $x \in \mathcal{C}^o(E)$ and $y \in \mathcal{C}^o(E')$

$$y \subseteq f(x) \Rightarrow \exists z \in \mathcal{C}^o(E). z \subseteq x \text{ and } fz = y.$$

The configuration z is necessarily unique by the local injectivity of f . (Again, the class of rigid maps would be unaffected if we allow all configurations in the definition above.)

Proposition 2.1 *A total map $f : E \rightarrow E'$ of event structures is rigid iff f preserves causal dependency, i.e., if $e' \leq e$ in E then $f(e') \leq f(e)$ in E' .*

Proof. “*If*”: Total maps reflect causal dependency. So, if f preserves causal dependency, then for any configuration x of E , the bijection $f : x \rightarrow fx$ preserves and reflects causal dependency. Hence for any subconfiguration y of fx , the bijection restricts to a bijection $f : z \rightarrow y$ with z a down-closed subset of x . But then z must be a configuration of E . “*Only if*”: Let $e \in E$. Then $[f(e)] \subseteq f[e]$. Hence, as f is rigid, there is a subconfiguration z of $[e]$ such that $fz = [f(e)]$. By local injectivity, $e \in z$, so $z = [e]$. Hence $f[e] = [f(e)]$. It follows that if $e' \leq e$ then $f(e') \leq f(e)$. \square

A rigid map of event structures preserves the causal dependency relation “rigidly,” so that the causal dependency relation on the image fx is a copy of that on a configuration x of E —in this sense f is a local isomorphism. This is not so for general maps where x may be augmented with extra causal dependency over that on fx . (Special forms of rigid maps appeared as *rigid embeddings* in Kahn and Plotkin’s work on concrete domains [12].)

Note that for any partial map of event structures $f : A \rightarrow B$ there is a least event structure B_0 , its *image*, included in B , with the inclusion forming a rigid map $j : B_0 \hookrightarrow B$, so that f factors as $j \circ f_0$ for map f_0 —the map f_0 will be total or rigid if the original map f is total or rigid, respectively. Construct B_0 to comprise the events fA with causal dependency inherited from B and with a finite subset of B_0 consistent iff it is the image of a consistent set in A .

Definition 2.2 Write \mathcal{E} for the category of event structures with total maps. Write \mathcal{E}_r and \mathcal{E}_p for the categories of event structures with rigid and partial maps, respectively. (We shall concentrate on total maps, and unless it is said otherwise a map of event structures will be a total map.)

The categories \mathcal{E} and \mathcal{E}_p are well-known and have a long history. The category \mathcal{E}_p is especially relevant to the semantics of process languages such as CCS and CSP based on event synchronisation, in particular its product is fundamental to the semantics of parallel compositions [23,21]. The category of rigid maps \mathcal{E}_r has been less studied, and its maps appear overly restrictive at first sight—after all projections from parallel compositions to their components are rarely rigid as a parallel composition most often imposes additional causal dependency on events of its components. However, as we shall see there is a strong case for the primary nature of rigid maps; other maps of event structures including total and partial maps can be obtained as rigid maps in Kleisli categories w.r.t. monads on event structures with rigid maps (though to obtain partial maps in this way we shall first have to extend event structures with symmetry). The primacy of rigid maps and the need for symmetry are glimpsed in the following propositions and their proofs.

Proposition 2.3 *The inclusion functor $\mathcal{E}_r \hookrightarrow \mathcal{E}$ has a right adjoint. The category \mathcal{E} is isomorphic to the Kleisli category of the monad for the adjunction.*

Proof. The right adjoint’s action on objects is given as follows. Let B be an event structure. For $x \in \mathcal{C}(B)$, an *augmentation* of x is a partial order (x, α) where $\forall b, b' \in x. b \leq_B b' \Rightarrow b \alpha b'$. We can regard such augmentations as elementary event structures in which all subsets of events are consistent. Order all augmentations

by taking $(x, \alpha) \sqsubseteq (x', \alpha')$ iff $x \subseteq x'$ and the inclusion $i : x \hookrightarrow x'$ is a rigid map $i : (x, \alpha) \rightarrow (x', \alpha')$. Augmentations under \sqsubseteq form a prime algebraic domain—see Appendix B; the complete primes are precisely the augmentations with a top element. Define $aug(B)$ to be its associated event structure.

There is an obvious total map of event structures $\epsilon_B : aug(B) \rightarrow B$ taking a complete prime to the event which is its top element. It can be checked that post-composition by ϵ_B yields a bijection

$$\epsilon_B \circ - : \mathcal{E}_r(A, aug(B)) \cong \mathcal{E}(A, B) .$$

Hence aug extends to a right adjoint to the inclusion $\mathcal{E}_r \hookrightarrow \mathcal{E}$.

Write aug also for the monad induced by the adjunction and $Kl(aug)$ for its Kleisli category. Under the bijection of the adjunction

$$Kl(aug)(A, B) =_{\text{def}} \mathcal{E}_r(A, aug(B)) \cong \mathcal{E}(A, B) .$$

The categories $Kl(aug)$ and \mathcal{E} share the same objects, and so are isomorphic. \square

Can the category \mathcal{E}_p , with partial maps, also be obtained as a Kleisli category from a monad on \mathcal{E}_r or \mathcal{E} ? A first thought would be a ‘monad’ $(-)_*$ adjoining an ‘undefined event’ $*$ to event structures and to represent partial maps of event structures from A to B as total maps from A to B_* . But then we would need to send several consistent events to the same ‘undefined event’ $*$ —which is not possible for maps of event structures. Some enlargement of event structures and their maps is needed before we can realize partial maps via a Kleisli construction.

Proposition 2.4 *The categories \mathcal{E}_r , \mathcal{E} and \mathcal{E}_p have products and pullbacks, though in the case of total and rigid maps there are no terminal objects.*

Proof. The nature of products in \mathcal{E} and \mathcal{E}_p is known from [23] and from these the existence of pullbacks follows directly—see Appendix C. The existence of products and pullbacks in \mathcal{E}_r follows from Appendix C.

The category \mathcal{E}_p has the empty event structure as terminal object. To see that \mathcal{E}_r and \mathcal{E} fail to have a terminal object, consider maps from 2 , the event structure comprising two concurrent events, to any putative terminal object \top . By the properties of maps, rigid or total, their image of 2 in \top must be two concurrent events e_0 and e_1 for which $[e_0] = \{e_0\}$ and $[e_1] = \{e_1\}$. But then there would be at least two maps from 1 , the event structure with a single event, to \top —contradicting uniqueness. (Later, with the introduction of symmetry, there will be a biterminal object, and then these two maps will be equal *up to symmetry*.) \square

Open maps

In defining symmetries on event structures we will make use of open maps [11]. Open maps are a generalisation of functional bisimulations, known from transition systems. They are specified as those maps in a category which satisfy a path-lifting property w.r.t. a chosen subcategory of paths. Here we take the subcategory of

paths to be the full subcategory of finite elementary event structures (*i.e.*, finite event structures in which all subsets are consistent).

W.r.t. a category of event structures (*i.e.*, \mathcal{E} , \mathcal{E}_r or \mathcal{E}_p), say a map $h : A \rightarrow B$, between event structures A and B , is *open* iff for all maps $j : p \rightarrow q$ between finite elementary event structures, any commuting square

$$\begin{array}{ccc} p & \xrightarrow{x} & A \\ j \downarrow & & \downarrow h \\ q & \xrightarrow{y} & B \end{array}$$

can be split into two commuting triangles

$$\begin{array}{ccc} p & \xrightarrow{x} & A \\ j \downarrow & \dashrightarrow z & \downarrow h \\ q & \xrightarrow{y} & B. \end{array}$$

That the square commutes means that the path $h \circ x$ in B can be extended via j to a path y in B . That the two triangles commute means that the path x can be extended via j to a path z in A which matches y .

Open maps compose and so form a subcategory, are preserved under pullbacks, and the product of open maps is open. All these facts follow purely diagrammatically [11]. W.r.t. any of the categories of event structures, we can characterise open maps as rigid maps with an extension property:

Proposition 2.5 *In any of the categories \mathcal{E} , \mathcal{E}_r and \mathcal{E}_p , a map $h : A \rightarrow B$ of event structures is open iff h is rigid and satisfies*

$$\forall x \in \mathcal{C}^o(A), y' \in \mathcal{C}^o(B). hx \subseteq y' \Rightarrow \exists x' \in \mathcal{C}(E). x \subseteq x' \ \& \ hx' = y' .$$

Proof. We show the result for the most general category \mathcal{E}_p . (The proofs for the other categories \mathcal{E} and \mathcal{E}_r use the same ideas.)

“ \Rightarrow ”: Assume h is open in \mathcal{E}_p . For $x \in \mathcal{C}^o(A)$, consider the commuting square

$$\begin{array}{ccc} x & \hookrightarrow & A \\ h_x \downarrow & & \downarrow h \\ hx & \hookrightarrow & B \end{array}$$

got by restricting h to the configuration x —both x and hx are regarded as elementary event structures with causal dependency inherited from A and B , respectively. Because h is open, we can factor the square into two commuting triangles:

$$\begin{array}{ccc} x & \hookrightarrow & A \\ h_x \downarrow & \dashrightarrow & \downarrow h \\ hx & \hookrightarrow & B. \end{array}$$

Because the upper triangle commutes, the restriction h_x of h to x and the ‘diagonal’ (dotted) map must be total. As the ‘diagonal’ reflects causal dependency, h_x must preserve causal dependency for events in x . For any two events $a \leq a'$ in A the finite configuration $[e']$ contains them both. Hence h preserves causal dependency, so is rigid.

Suppose $hx \subseteq y'$ for $x \in \mathcal{C}^o(A)$ and $y' \in \mathcal{C}^o(B)$. Again, regard x , hx and y' as elementary event structures with causal dependency inherited from their ambient event structures. We obtain the commuting square

$$\begin{array}{ccc}
 x & \hookrightarrow & A \\
 \downarrow & & \downarrow \\
 hx & & h \\
 \downarrow & & \downarrow \\
 y' & \hookrightarrow & B
 \end{array}$$

where we have also added its factorization into two commuting triangles due to h being open. Commutativity of the triangles, yields (as the image of the dotted ‘diagonal’) a finite configuration x' of A with $x \subseteq x'$ and $hx' = y'$.

“ \Leftarrow ”: Assume h is rigid and satisfies the extension property. Suppose for elementary event structures p and q that

$$\begin{array}{ccc}
 p & \longrightarrow & A \\
 \downarrow & & \downarrow \\
 q & \longrightarrow & B
 \end{array}$$

commutes in \mathcal{E}_p . Factoring through the images x of p and y of q we obtain two commuting squares

$$\begin{array}{ccccc}
 p & \twoheadrightarrow & x & \hookrightarrow & A \\
 \downarrow & & \downarrow & & \downarrow \\
 & & hx & & h \\
 \downarrow & & \downarrow & & \downarrow \\
 q & \twoheadrightarrow & y & \hookrightarrow & B
 \end{array}$$

By the extension property, there is $x' \in \mathcal{C}^o(A)$ such that $hx' = y$. Because h is rigid, there is a (dotted) ‘diagonal’ map, sending y to x' , which breaks the rightmost square into two commuting triangles. Composing the diagonal with the map $q \rightarrow y$ splits the outer commuting square in the way required for h to be open. \square

3 Event structures with symmetry

We shall present a general definition of symmetry, concentrating on the category \mathcal{E} of event structures with total maps. This category has (binary) products and pullbacks (though no terminal object) and supports a notion of open map. For the definition of symmetry we are about to give this is all we require.

A symmetry on an event structure should specify which events are similar in such a way that similar events have similar pasts and futures. This is captured, somewhat abstractly, by the following definition.

Definition 3.1 An *event structure with symmetry* (E, l, r) comprises an event structure E together with *open* maps $l : S \rightarrow E$ and $r : S \rightarrow E$ from a common event structure S such that the map $\langle l, r \rangle : S \rightarrow E \times E$ is an equivalence relation (*i.e.*, the map $\langle l, r \rangle$ is monic—equivalently, l, r are jointly monic—and satisfies the standard diagrammatic properties of reflexivity, symmetry and transitivity [10]. See Appendix A).

A bisimulation is given by a span of open maps [11], in the case of the above definition by the pair of open maps l and r . So the definition expresses a symmetry on an event structure as a bisimulation equivalence. The definition has the advantage of being abstract in that it readily makes sense for any category with binary products and pullbacks for which there is a sensible choice of paths in order to define open maps. It is sensible for the categories of event structures with rigid and partial maps, for stable families, transition systems, trace languages and Petri nets [21], because these categories also have products, pullbacks and open maps; both categories of event structures with rigid and partial maps would have the same class of open maps and so lead to precisely the same event structures with symmetry as objects. We shall mainly concentrate on the category with total maps to connect directly with the particular examples we shall treat here.⁴

For the specific model of event structures there is an alternative way to present a symmetry. We can express a symmetry $l, r : S \rightarrow E$ on an event structure E equivalently as a relation of similarity between its finite configurations. More precisely, two finite configurations x, y of E are related by a bijection $\theta_z =_{\text{def}} \{(l(s), r(s)) \mid s \in z\}$ if they arise as images $x = lz$ and $y = rz$ of a common finite configuration z of S ; because l and r are locally injective θ_z is a bijection between x and y . Because l and r are rigid the bijection is an order isomorphism between x and y with the order of causal dependency inherited from E . In this way a symmetry on E will determine an *isomorphism family* expressing when and how two finite configurations are similar, or symmetric, in the sense that one can replace the other. As expected, such similarity forms an equivalence relation, and if two configurations are similar then so are their pasts (restrictions to subconfigurations) and futures (extensions to larger configurations).

Definition 3.2 An *isomorphism family* of an event structure E consists of a family \mathbb{S} of bijections

$$\theta : x \cong y$$

between pairs of finite configurations of E such that:

(i) the identities $\text{id}_x : x \cong x$ are in \mathbb{S} for all $x \in \mathcal{C}^o(E)$; if $\theta : x \cong y$ is in \mathbb{S} , then so is the inverse $\theta^{-1} : y \cong x$; and if $\theta : x \cong y$ and $\varphi : y \cong z$ are in \mathbb{S} , then so is their

⁴ As we shall see, there is a strong case for regarding rigid maps as *the* fundamental maps of event structures, in that other maps on event structures can then ultimately be obtained as Kleisli maps w.r.t. suitable pseudo monads once we have introduced symmetry.

composition $\varphi \circ \theta : x \cong z$.

(ii) for $\theta : x \cong y$ in \mathbb{S} whenever $x' \subseteq x$ with $x' \in \mathcal{C}(E)$, then there is a (necessarily unique) $y' \in \mathcal{C}(E)$ with $y' \subseteq y$ such that the restriction of θ to $\theta' : x' \cong y'$ is in \mathbb{S} .

(iii) for $\theta : x \cong y$ in \mathbb{S} whenever $x \subseteq x'$ for $x' \in \mathcal{C}^o(E)$, then there is an extension of θ to $\theta' : x' \cong y'$ in \mathbb{S} for some (not necessarily unique) $y' \in \mathcal{C}^o(E)$ with $y \subseteq y'$.

[Note that (i) implies that the converse forms of (ii) and (iii) also hold. Note too that (ii) implies that the bijections in the family \mathbb{S} respect the partial order of causal dependency on configurations inherited from E ; the bijections in an isomorphism family are isomorphisms between the configurations regarded as elementary event structures.]

We shall use the following simple fact about rigid maps in showing the correspondence between symmetries on event structures and isomorphism families.

Lemma 3.3 *A rigid map of event structures $h : A \rightarrow B$ which is injective and surjective as a function $h : \mathcal{C}^o(A) \rightarrow \mathcal{C}^o(B)$, sending a configuration to its image, is an isomorphism of event structures.*

Proof. Consider the function $h : \mathcal{C}^o(A) \rightarrow \mathcal{C}^o(B)$ induced between configurations. For $x, y \in \mathcal{C}^o(A)$,

$$x \subseteq y \Leftrightarrow hx \subseteq hy .$$

“ \Rightarrow ”: Obvious. “ \Leftarrow ”: If $hx \subseteq hy$, then by rigidity, there is a configuration $y' \subseteq y$ such that $hy' = hx$. But by injectivity $x = y'$, so $x \subseteq y$.

Being surjective, h is a bijection between finite configurations which preserves and reflects inclusion. Being rigid, h preserves and reflects prime configurations. Because the event structures can be recovered via the primes—Theorem B.1, this entails that h is an isomorphism of event structures. \square

Isomorphism families are really symmetries on stable families (as we will spell out later in Theorem 5.2). Accordingly the proof of the following key theorem rests on the coreflection between the categories of event structures and stable families described in Appendix C.

Theorem 3.4 *Let E be an event structure.*

(i) *A symmetry $l, r : S \rightarrow E$ determines an isomorphism family \mathbb{S} : defining $\theta_z = \{(l(s), r(s)) \mid s \in z\}$ for z a finite configuration of S , yields a bijection $\theta_z : lz \cong rz$; the family \mathbb{S} consisting of all bijections $\theta_z : lz \cong rz$, for z a finite configuration of S .*

(ii) *An isomorphism family \mathbb{S} of E determines a symmetry $l, r : S \rightarrow E$: the family \mathbb{S} forms a stable family; the event structure S is obtained as $\text{Pr}(\mathbb{S})$ for which the events are primes $[(e_1, e_2)]_\theta$ for θ in \mathbb{S} and $(e_1, e_2) \in \theta$; the maps l and r send a prime $[(e_1, e_2)]_\theta$ to e_1 and e_2 respectively.*

The operations of (i) and (ii) are mutually inverse (regarding relations as sub-objects). Performing (ii) then (i) returns the original isomorphism family. Starting from a symmetry $l, r : S \rightarrow E$, performing (i) then (ii) produces a symmetry $l', r' : \text{Pr}(\mathbb{S}) \rightarrow E$, via the isomorphism family \mathbb{S} of (i). There is an isomorphism of

event structures $h : S \rightarrow \text{Pr}(\mathbb{S})$ given by $h(s) = \{(l(s'), r(s')) \mid s' \leq s\}$, for $s \in S$. The maps l', r' satisfy $l'(h(s)) = l(s)$ and $r'(h(s)) = r(s)$, for $s \in S$.

Proof. (i) Let $l, r : S \rightarrow E$ be a symmetry. That θ_z is indeed a bijection from $l z$ to $r z$ for $z \in \mathcal{C}^o(S)$, follows directly from l and r being maps of event structures. That the collection of all such bijections satisfies property (i) of Definition 3.2 is a routine consequence of $\langle l, r \rangle$ forming an equivalence relation. The remaining properties, (ii) and (iii), follow from l and r being open.

(ii) Let \mathbb{S} be an isomorphism family of E . That \mathbb{S} forms a stable family is a consequence of property (ii) of the isomorphism family, using the fact that $\mathcal{C}^o(E)$ is itself a stable family. Note that inclusion of events induces a rigid map of stable families $\mathbb{S} \hookrightarrow \mathcal{C}^o(E) \times \mathcal{C}^o(E)$ to the product $\mathcal{C}^o(E) \times \mathcal{C}^o(E)$ in \mathcal{Fam} , again by property (ii) of the isomorphism family \mathbb{S} .

The rigid inclusion map of stable families $j : \mathbb{S} \hookrightarrow \mathcal{C}^o(E) \times \mathcal{C}^o(E)$ yields a rigid inclusion map of event structures $\text{Pr } j : \text{Pr}(\mathbb{S}) \hookrightarrow \text{Pr}(\mathcal{C}^o(E) \times \mathcal{C}^o(E))$, i.e. $\text{Pr } j; \text{Pr}(\mathbb{S}) \hookrightarrow E \times E$ —as in Appendix C.1, we take the product $E \times E$ in \mathcal{E} to be $\text{Pr}(\mathcal{C}^o(E) \times \mathcal{C}^o(E))$. Composing with the projections we obtain the maps $l, r : \text{Pr}(\mathbb{S}) \rightarrow E$: they map a prime $[(e_1, e_2)]_\theta$ to e_1 and e_2 respectively. They are open by properties (ii) and (iii) of the isomorphism family \mathbb{S} .

We check that performing (ii) then (i) returns the original isomorphism family. Starting with an isomorphism family \mathbb{S} via (ii) we obtain the symmetry $l, r : \text{Pr}(\mathbb{S}) \rightarrow E$ where $l([(e_1, e_2)]_\theta) = e_1$ and $r([(e_1, e_2)]_\theta) = e_2$. Then via (i) we obtain an isomorphism family \mathbb{S}' consisting of all bijections ϕ for which

$$\phi = \{(l(s), r(s)) \mid s \in z\}$$

for some $z \in \mathcal{C}^o(\text{Pr}(\mathbb{S}))$. But the configurations $\mathcal{C}^o(\text{Pr}(\mathbb{S}))$ are precisely those subsets z for which

$$z = \{[(e_1, e_2)]_\theta \mid (e_1, e_2) \in \theta\}$$

for some $\theta \in \mathbb{S}$ —see Appendix, Theorem C.4. It follows that $\mathbb{S} = \mathbb{S}'$.

On the other hand, performing (i) then (ii), starting from a symmetry $l, r : S \rightarrow E$, produces another symmetry $l', r' : \text{Pr}(\mathbb{S}) \rightarrow E$, from \mathbb{S} , the isomorphism family described in (i). However, $\langle l, r \rangle$ and $\langle l', r' \rangle$ are the same relation in the sense of representing the same subobject of $E \times E$; they do so via the isomorphism $h : S \cong \text{Pr}(\mathbb{S})$ given by $s \mapsto \{(l(s'), r(s')) \mid s' \leq s\}$. To see this argue as follows.

From the original symmetry $l, r : S \rightarrow E$, we obtain the following commuting

diagram in \mathcal{Fam} :

$$\begin{array}{ccc}
 & \mathcal{C}^o(S) & \\
 & \downarrow d & \\
 \mathcal{C}^o(E) & \xleftarrow{\pi_1} \mathcal{C}^o(E) \times \mathcal{C}^o(E) \xrightarrow{\pi_2} & \mathcal{C}^o(E)
 \end{array}$$

where we have factored the mediating map $d : \mathcal{C}^o(S) \rightarrow \mathcal{C}^o(E) \times \mathcal{C}^o(E)$, for which $d(s) = (l(s), r(s))$, through its image, the isomorphism family \mathbb{S} .

Applying the functor Pr we obtain the commuting diagram in \mathcal{E}

$$\begin{array}{ccccc}
 & & S & & \\
 & & \downarrow \eta_S & & \\
 & & \text{Pr } \mathcal{C}^o(S) & & \\
 & \swarrow l & \downarrow \text{Pr } d & \searrow r & \\
 & \text{Pr } \mathcal{C}^o(E) & \downarrow \text{Pr } \mathbb{S} & \text{Pr } \mathcal{C}^o(E) & \\
 \eta_E \swarrow & \xleftarrow{\text{Pr } \pi_1} E \times E \xrightarrow{\text{Pr } \pi_2} & \eta_E \searrow & & \\
 E & & E & & E
 \end{array}$$

where we have also added the (dotted) naturality ‘squares’ associated with the natural isomorphism η , the unit of the adjunction between event structures \mathcal{E} and stable families \mathcal{Fam} . Bearing in mind the definition, from Appendix C.1, of the product $E \times E, p_1, p_2$ in \mathcal{E} we can simplify this to the commuting diagram

$$\begin{array}{ccc}
 & S & \\
 & \downarrow h & \\
 & \text{Pr } (\mathbb{S}) & \\
 \swarrow l & & \searrow r \\
 E & \xleftarrow{p_1} E \times E \xrightarrow{p_2} & E
 \end{array} \tag{\dagger}$$

where we have defined $h =_{\text{def}} (\text{Pr } d) \circ \eta_S$. Clearly the mediating map to the product must equal $\langle l, r \rangle$, *i.e.*

$$\begin{array}{ccc}
 S & \xrightarrow{h} & \text{Pr } (\mathbb{S}) \\
 \searrow \langle l, r \rangle & & \downarrow \\
 & & E \times E
 \end{array} \tag{\ddagger}$$

commutes.

We must have h monic as $\langle l, r \rangle$ is monic. Moreover l and r are rigid so $\langle l, r \rangle$ is rigid—Appendix C.1—ensuring that h is rigid too. It is not hard to see that $\text{Pr } d$ is surjective as a map $\text{Pr } d : \mathcal{C}^o(\text{Pr } \mathcal{C}^o(S)) \rightarrow \text{Pr } (\mathbb{S})$ on configurations—this follows because d is surjective on configurations by its definition. Hence h is also surjective

on configurations. The rigid map h is now injective (being monic) and surjective on configurations, so by Lemma 3.3 it is an isomorphism of event structures $h : S \cong \text{Pr}(S)$. Unwrapping the definition of h as $(\text{Pr}d) \circ \eta_S$ we see that $h(s) = \{(l(s'), r(s')) \mid s' \leq s\}$ for $s \in S$.

By definition the symmetry maps $l', r' : \text{Pr}(S) \rightarrow E$ are given by projections, so filling in the diagram (†) we obtain the commuting diagram

$$\begin{array}{ccccc}
 & & S & & \\
 & & \downarrow h & & \\
 & l & \text{Pr}(S) & r & \\
 & \swarrow & \downarrow & \searrow & \\
 E & \xleftarrow{p_1} & E \times E & \xrightarrow{p_2} & E \\
 & \swarrow l' & \downarrow & \searrow r' & \\
 & & E & &
 \end{array}$$

from which $l'(h(s)) = l(s)$ and $r'(h(s)) = r(s)$ for all $s \in S$. The inclusion $\text{Pr}(S) \hookrightarrow E \times E$ must equal $\langle l', r' \rangle$. So by (†), as h is an isomorphism, the two relations $\langle l, r \rangle$ and $\langle l', r' \rangle$ are equal (as subobjects of the product). \square

Through the addition of symmetry event structures can represent a much richer class of ‘path categories’ [4] than mere partial orders. The finite configurations of an event structure with symmetry can be extended by inclusion or rearranged bijectively under an isomorphism allowed by the symmetry. In this way an event structure with symmetry determines, in general, a *category* of finite configurations with maps obtained by repeatedly composing the inclusions and allowed isomorphisms. By property (ii) in Definition 3.2 any such map factors uniquely as an isomorphism of the symmetry followed by an inclusion. While by property (iii) any such map factors (not necessarily uniquely) as an inclusion followed by an isomorphism of the symmetry.

Example 3.5 Any event structure E can be identified with the event structure with the identity symmetry $(E, \text{id}_E, \text{id}_E)$. Its isomorphism family consists of all identities $\text{id}_x : x \cong x$ on finite configurations $x \in \mathcal{C}(E)$.

Example 3.6 Identify the natural numbers ω with the event structure with events ω , trivial causal dependency given by the identity relation and in which all finite subsets of events are in the consistency relation. Define S to be the product of event structures $\omega \times \omega$ in \mathcal{E} ; the product comprises events all pairs $(i, j) \in \omega \times \omega$ with trivial causal dependency, and consistency relation consisting of all finite subsets of $\omega \times \omega$ which are bijective (so we take two distinct pairs (i, j) and (i', j') to be in conflict iff $i = i'$ or $j = j'$.) Define l and r to be the projections $l : S \rightarrow E$ and $r : S \rightarrow E$. Then $\varpi =_{\text{def}} (\omega, l, r)$ forms an event structure with symmetry. The corresponding isomorphism family in this case coincides with all finite bijections between finite subsets of ω . Any finite subset of events of ϖ is similar to any other. Of course, an analogous construction works for any countable, possibly finite, set.

Example 3.7 Let $E = (E, l : S \rightarrow E, r : S \rightarrow E)$ be an event structure with symmetry. Define an event structure with symmetry $!E = (E!, l! : S! \rightarrow E!, r! : S!$

$S_!$ comprising ω similar copies of E as follows. The event structure $E_!$ has the set of events $\omega \times E$ with causal dependency

$$(i, e) \leq_! (i', e') \text{ iff } i = i' \ \& \ e \leq_E e'$$

and consistency relation

$$C \in \text{Con}_! \text{ iff } C \text{ is finite \& } \forall i \in \omega. \{e \mid (i, e) \in C\} \in \text{Con}_E .$$

The symmetry $S_!$ has events $\omega \times \omega \times S$ with causal dependency

$$(i, j, s) \leq_{S_!} (i', j', s') \text{ iff } i = i' \ \& \ j = j' \ \& \ s \leq_S s' .$$

A finite subset $C \subseteq S_!$ is in the consistency relation $\text{Con}_{S_!}$ iff

$$\{(i, j) \mid \exists s. (i, j, s) \in C\} \text{ is bijective \& } \forall i, j \in \omega. \{s \mid (i, j, s) \in C\} \in \text{Con}_S .$$

Define $l(i, j, s) = (i, l_S(s))$ and $r(i, j, s) = (j, r_S(s))$ for $i, j \in \omega, s \in S$.

The finite configurations of $E_!$ correspond to tuples (or indexed families) $\langle x_i \rangle_{i \in I}$ of nonempty-finite configurations $x_i \in \mathcal{C}(E)$ indexed by $i \in I$, where I a finite subset of ω . With this view of the configurations of $E_!$, the isomorphism family corresponding to $S_!$ specifies isomorphisms between tuples

$$(\sigma, \langle \theta_i \rangle_{i \in I}) : \langle x_i \rangle_{i \in I} \cong \langle y_j \rangle_{j \in J}$$

consisting of a bijection between indices $\sigma : I \cong J$ together with $\theta_i : x_i \cong y_{\sigma(i)}$ from the isomorphism family of S , for all $i \in I$.

The event structure with symmetry ϖ reappears as the special case $!1$, where 1 is the event structure with a single event.

We conclude this section with a general method for constructing symmetries, though one we will not use further in this paper. Just as there is a least symmetry on an event structure, *viz.* the identity symmetry, so is there a greatest. Moreover any bisimulation on an event structure generates a symmetry on it. We take a *bisimulation* on an event structure A to be a pair of open maps $l, r : R \rightarrow A$ from an event structure R for which $\langle l, r \rangle$ is monic. In general we might specify a bisimulation on an event structure just by a pair of open maps from a common event structure, and not insist that the pair is monic. But here, no real generality is lost as such a pair of open maps on event structures will always factor through its image, a bisimulation with monicity. The proof proceeds most easily by first establishing an analogous property for isomorphism families. We define a *bisimulation family* to be a family of bijections between finite configurations of A which satisfy (ii) and (iii) in Definition 3.2.

Proposition 3.8 *Let A be an event structure.*

(i) *For any bisimulation family \mathcal{R} on A there is a least isomorphism family \mathbb{S} for which $\mathcal{R} \subseteq \mathbb{S}$.*

(ii) For any bisimulation $\langle l_0, r_0 \rangle : R \rightarrow A$ there is a least symmetry $\langle l, r \rangle : S \rightarrow A$ (understood as a subobject) for which R is a subobject of S .

There is a greatest symmetry on A .

Proof. (i) The family \mathcal{R} can be inductively closed under identities, symmetry and transitivity, while maintaining the properties of a bisimulation family, to form an isomorphism family \mathbb{S} . The inductive construction of \mathbb{S} ensures that it is the least isomorphism family including \mathcal{R} .

(ii) The correspondence between symmetries and isomorphism families of Theorem 3.4 extends to a correspondence between bisimulations and bisimulation families—by copying the proof there. A bisimulation R corresponds to a bisimulation family \mathcal{R} . The least isomorphism family \mathbb{S} including \mathcal{R} corresponds to a least bisimulation R .

Bisimulation families are closed under unions. Hence there is a maximum such family, necessarily the greatest isomorphism family of A , as closure under identities, symmetry and transitivity maintains the properties of a bisimulation family. The greatest isomorphism family determines the greatest symmetry on A . \square

4 Maps preserving symmetry

Maps between event structures with symmetry are defined as maps between event structures which preserve symmetry. Let (A, l_A, r_A) and (B, l_B, r_B) be event structures with symmetry. A map $f : (A, l_A, r_A) \rightarrow (B, l_B, r_B)$ is a map of event structures $f : A \rightarrow B$ such that there is a (necessarily unique) map of event structures $h : S_A \rightarrow S_B$ ensuring

$$\langle l_B, r_B \rangle \circ h = (f \times f) \circ \langle l_A, r_A \rangle .$$

Here we are adopting a convention to be used throughout the paper: when otherwise unspecified we shall assume that A an event structure with symmetry has open maps described as $l_A, r_A : S_A \rightarrow A$ from an event structure S_A ; we shall also often understand Con_A and \leq_A as its consistency and causal dependency relations.

Note the obviously equivalent characterization of a map $f : (A, l_A, r_A) \rightarrow (B, l_B, r_B)$ preserving symmetry as a map of the underlying event structures $f : A \rightarrow B$ for which there is a (necessarily unique) map $h : S_A \rightarrow S_B$ making

$$\begin{array}{ccccc} A & \xleftarrow{l_A} & S_A & \xrightarrow{r_A} & A \\ f \downarrow & & \downarrow h & & \downarrow f \\ B & \xleftarrow{l_B} & S_B & \xrightarrow{r_B} & B \end{array}$$

commute.

Maps between event structures with symmetry compose as maps of event structures and share the same identity maps.

Definition 4.1 We define \mathcal{S} to be category of event structures with symmetry.

We can characterize when maps of event structures preserve symmetry in terms of isomorphism families. A map preserving symmetry should behave as a functor both w.r.t. the inclusion between finite configurations and the isomorphisms of the symmetry.

Proposition 4.2 *A map of event structures $f : A \rightarrow B$ is a map $f : (A, l_A, r_A) \rightarrow (B, l_B, r_B)$ of event structures with symmetry iff whenever $\theta : x \cong y$ is in the isomorphism family of A then $f\theta : fx \cong fy$ is in the isomorphism family of B , where $f\theta =_{\text{def}} \{(f(e_1), f(e_2)) \mid (e_1, e_2) \in \theta\}$.*

Proof. ‘*Only if*’: Assume $f : (A, l_A, r_A) \rightarrow (B, l_B, r_B)$ is a map of event structures with symmetry. By definition, $\langle l_B, r_B \rangle \circ h = (f \times f) \circ \langle l_A, r_A \rangle$ for some map $h : S_A \rightarrow S_B$. We thus have the equations $fl_A = l_Bh$ and $fr_A = r_Bh$. Any bijection $\theta : x \cong y$ in the isomorphism family of A is obtained as $\theta = \{(l_A(s), r_A(s)) \mid s \in z\}$ for some $z \in \mathcal{C}^o(S_A)$. The configuration $hz \in \mathcal{C}^o(S_B)$ determines the bijection $\{(l_B(s), r_B(s)) \mid s \in hz\}$ in the isomorphism family of B . But this bijection coincides with $f\theta$ by the equations above.

‘*If*’: Suppose f respects the isomorphism families \mathbb{S}_A of A and \mathbb{S}_B of B as described in the proposition. Then diagrammatically in \mathcal{Fam} , the category of stable families,

$$\begin{array}{ccc} \mathbb{S}_A & \xrightarrow{h} & \mathbb{S}_B \\ \downarrow & & \downarrow \\ \mathcal{C}^o(A) \times \mathcal{C}^o(A) & \xrightarrow{f \times f} & \mathcal{C}^o(B) \times \mathcal{C}^o(B) \end{array}$$

commutes for some unique map h of stable families (the downwards maps are inclusions). Applying Pr we obtain the commuting diagram

$$\begin{array}{ccccc} S_A & \cong & \text{Pr}(\mathbb{S}_A) & \xrightarrow{\text{Pr}h} & \text{Pr}(\mathbb{S}_B) & \cong & S_B \\ & \swarrow \langle l_A, r_A \rangle & \downarrow & & \downarrow & \swarrow \langle l_B, r_B \rangle & \\ & & A \times A & \xrightarrow{f \times f} & B \times B & & \end{array}$$

where we have also added the isomorphisms with the original symmetries. Hence f is a map of event structures with symmetry. \square

We explore properties of the category \mathcal{E} . It is more fully described as a category enriched in the category of equivalence relations and so, because equivalence relations are a degenerate form of category, as a 2-category in which the 2-cells are instances of the equivalence \sim . This view informs the constructions in \mathcal{E} which are often very simple examples of the (pseudo- and bi-) constructions of 2-categories.

Definition 4.3 Let $f, g : (A, l_A, r_A) \rightarrow (B, l_B, r_B)$ be maps of event structures with symmetry (A, l_A, r_A) and (B, l_B, r_B) . Define $f \sim g$ iff there is a (necessarily unique) map of event structures $h : A \rightarrow S_B$ such that

$$\langle f, g \rangle = \langle l_B, r_B \rangle \circ h .$$

Note the obviously equivalent way to express $f \sim g$, through the existence of a (necessarily unique) map h such that the following diagram commutes:

$$\begin{array}{ccc}
 & A & \\
 f \swarrow & & \searrow g \\
 B & \xleftarrow{l} S_B \xrightarrow{r} & B \\
 & \downarrow h \downarrow & \\
 & &
 \end{array}$$

Straightforward diagrammatic proofs show:

Proposition 4.4 *The relation \sim is an equivalence relation on maps $\mathcal{SE}(A, B)$ between event structures with symmetry A and B . The relation \sim respects composition in the sense that if $f \sim g$ then $h \circ f \circ k \sim h \circ g \circ k$, for composable maps h and k .*

The category \mathcal{SE} is enriched in the category of equivalence relations (comprising equivalence relations with functions which preserve the equivalence).

We can characterize the equivalence of maps between event structures with symmetry in terms of isomorphism families which makes apparent how \sim is an instance of natural isomorphism between functors.

Proposition 4.5 *Let $f, g : (A, l_A, r_A) \rightarrow (B, l_B, r_B)$ be maps of event structures with symmetry. Then, $f \sim g$ iff $\theta_x : f x \cong g x$ is in the isomorphism family of (B, l_B, r_B) for all $x \in \mathcal{C}^o(A)$, where $\theta_x =_{\text{def}} \{(f(a), g(a)) \mid a \in x\}$.*

Proof. ‘Only if’: Assume $f \sim g$, i.e. the equations $f = l_B h$ and $g = r_B h$ hold for some map $h : A \rightarrow S_B$. Let $x \in \mathcal{C}^o(A)$. Then $h x \in \mathcal{C}^o(S_B)$ and this configuration determines the bijection $\{(l_B(s), r_B(s)) \mid s \in h x\}$ in the isomorphism family of B . But this bijection coincides with θ_x by the equations.

‘If’: Assume $\theta_x : f x \cong g x$ is in the isomorphism family \mathbb{S}_B of (B, l_B, r_B) for all $x \in \mathcal{C}^o(A)$. This may be expressed as the commuting triangle

$$\begin{array}{ccc}
 & & \mathbb{S}_B \\
 & \nearrow h & \downarrow \\
 \mathcal{C}^o(A) \times \mathcal{C}^o(A) & \xrightarrow{f \times g} & \mathcal{C}^o(B) \times \mathcal{C}^o(B)
 \end{array}$$

in \mathcal{Fam} . Applying Pr , moving to \mathcal{E} , we obtain the commuting diagram

$$\begin{array}{ccc}
 & \text{Pr}(\mathbb{S}_B) \cong S_B & \\
 \text{Pr } h \nearrow & \downarrow & \downarrow \langle l_B, r_B \rangle \\
 A \times A & \xrightarrow{f \times g} & B \times \dot{B}
 \end{array}$$

Hence $f \sim g$. □

Equivalence on maps yields an equivalence on objects:

Definition 4.6 Let A and B be event structures with symmetry. An *equivalence* from A to B is a pair of maps $f : A \rightarrow B$ and $g : B \rightarrow A$ such that $f \circ g \sim \text{id}_B$ and $g \circ f \sim \text{id}_A$; then we say A and B are equivalent and write $A \simeq B$.

5 General categories with symmetry

The procedure we have used to extend event structures with symmetry carries through for any category \mathcal{A} with (binary) products, pullbacks and a distinguished subcategory $\mathcal{O}_{\mathcal{A}}$ of ‘open’ maps which include all isomorphisms of \mathcal{A} and are such that the product of open maps is open and the pullback of an open map is open. We can then define an object with symmetry to be $(A, l, r : S \rightarrow A)$ consisting of an object A of \mathcal{A} and two open maps l, r which together make $\langle l, r \rangle$ an equivalence relation—just as for event structures with symmetry. Just as before we can define what it means for a map to preserve symmetry, and the equivalence relation \sim on homsets saying when two such maps are equivalent. In this way we produce a category \mathfrak{SA} enriched in the category of equivalence relations. In particular, we can form the categories of event structures with symmetry \mathfrak{SE}_r , based on rigid maps, and \mathfrak{SE}_p , based on partial maps (as well as corresponding categories of stable families—see Appendix C).

Now assume both $\mathcal{A}, \mathcal{O}_{\mathcal{A}}$ and $\mathcal{B}, \mathcal{O}_{\mathcal{B}}$ are categories with products and pullbacks, with respective subcategories of open maps. Certain functors between \mathcal{A} and \mathcal{B} straightforwardly induce functors between the enriched categories \mathfrak{SA} to \mathfrak{SB} of objects with symmetry.

Proposition 5.1 *Suppose a functor $F : \mathcal{A} \rightarrow \mathcal{B}$ preserves pullbacks, open maps, and has monic mediators for products in the sense that for all products $A \times A, \pi_1, \pi_2$ and $F(A) \times F(A), p_1, p_2$ the unique mediating map h in the commuting diagram*

$$\begin{array}{ccccc} & & F(A \times A) & & \\ & F(\pi_1) \swarrow & \downarrow h & \searrow F(\pi_2) & \\ F(A) & \xleftarrow{p_1} & F(A) \times F(A) & \xrightarrow{p_2} & F(A) \end{array}$$

is monic. Then F will induce a functor $\mathfrak{SF} : \mathfrak{SA} \rightarrow \mathfrak{SB}$ which takes an object with symmetry $(A, l : S \rightarrow A, r : S \rightarrow A)$ to an object with symmetry $(F(A), F(l), F(r))$ and a map $f : (A, l_A, r_A) \rightarrow (B, l_B, r_B)$ to $F(f)$. The functor \mathfrak{SF} preserves \sim on homsets.

Proof. Let $(A, l : S \rightarrow A, r : S \rightarrow A)$ be an object in \mathfrak{SA} . Consider the diagram

$$\begin{array}{ccccc} & & F(S) & & \\ & & \downarrow F\langle l, r \rangle & & \\ & Fl \swarrow & F(A \times A) & \searrow Fr & \\ & F(\pi_1) \swarrow & \downarrow h & \searrow F(\pi_2) & \\ F(A) & \xleftarrow{p_1} & F(A) \times F(A) & \xrightarrow{p_2} & F(A) . \end{array}$$

The map $\langle Fl, Fr \rangle$ equals the composition $h \circ F\langle l, r \rangle$ which is monic, because the mediating map h is monic, by assumption, and $F\langle l, r \rangle$ is monic, as F preserves pullbacks. It is a routine matter to check that $(F(A), Fl, Fr)$ is an object with symmetry and show the construction induces a \sim -respecting functor from \mathcal{SA} to \mathcal{SB} . \square

Proposition 5.1 will be quite useful. For now, observe that the right adjoint $\text{Pr} : \mathcal{Fam} \rightarrow \mathcal{E}$ from stable families to event structures preserves open maps, pullbacks and products, and so certainly meets the requirements of Proposition 5.1. It thus provides a functor $\mathcal{SPr} : \mathcal{SFam} \rightarrow \mathcal{E}$. The functor \mathcal{SPr} has a left adjoint, the now familiar construction of forming the isomorphism family of an event structure with symmetry. (The left adjoint to \mathcal{SPr} is not constructed via Proposition 5.1.)

Theorem 5.2 *The functor $\mathcal{SPr} : \mathcal{SFam} \rightarrow \mathcal{E}$ has a left adjoint $I : \mathcal{E} \rightarrow \mathcal{SFam}$ which preserves \sim on homsets. On objects, the functor I takes an event structure with symmetry (A, l, r) to the stable family with symmetry (\mathbb{S}, l', r') comprising \mathbb{S} the isomorphism family of (A, l, r) with symmetry maps $l', r' : \mathbb{S} \rightarrow \mathcal{C}^o(A)$ where $l'(a, a') = a$ and $r'(a, a') = a'$. The functor I takes a map $f : (A, l_A, r_A) \rightarrow (B, l_B, r_B)$ in \mathcal{E} to the map $f : \mathcal{C}^o(A) \rightarrow \mathcal{C}^o(B)$ sending a configuration $x \in \mathcal{C}^o(A)$ to the configuration $f x$.*

The adjunction $I \dashv \mathcal{SPr}$ has unit η , a natural isomorphism, and counit ϵ with components $\eta_A : A \rightarrow \text{Pr } \mathcal{C}^o(A)$, where $\eta_A(a) = [a]$ for $a \in A$, and $\epsilon_{\mathcal{F}} : \mathcal{C}^o \text{Pr}(\mathcal{F}) \rightarrow \mathcal{F}$, where $\epsilon_A([a]_x) = a$ for $x \in \mathcal{F}$ and $a \in x$ (they coincide with the unit and counit of the adjunction $\mathcal{C}^o \dashv \text{Pr}$). The bijection of the adjunction

$$\mathcal{E}(A, \mathcal{SPr}(\mathcal{F})) \cong \mathcal{SFam}(I(A), \mathcal{F}) ,$$

natural in A and \mathcal{F} , preserves and reflects \sim .

Proof. Given an event structure with symmetry (A, l, r) it is routine to check that its isomorphism family, with the two projections, is a stable family with symmetry. That a symmetry-preserving map between event structures with symmetry becomes a symmetry-preserving map between stable families with symmetry is a reformulation of the ‘only if’ direction of Proposition 4.2. The functoriality of I is obvious. That I preserves \sim follows directly from the ‘only if’ direction of Proposition 4.5.

We should check that the components of the unit and counit preserve symmetry.

For the unit we require w.r.t. an event structure with symmetry (A, l, r) that

$$\begin{array}{ccccc} A & \xleftarrow{l} & S & \xrightarrow{r} & A \\ \eta_A \downarrow & & \downarrow h & & \downarrow \eta_A \\ \text{Pr } \mathcal{C}^o(A) & \xleftarrow{\text{Pr } l'} & \text{Pr}(S) & \xrightarrow{\text{Pr } r'} & \text{Pr } \mathcal{C}^o(A) \end{array}$$

commutes, for some (unique) map h . But this is so for the isomorphism h of Theorem 3.4, for which $h(s) = \{(l(s'), r(s')) \mid s' \leq s\}$ when $s \in S$.

For the counit we require w.r.t. a stable family \mathcal{F} with symmetry $L, R : \mathbb{S} \rightarrow \mathcal{F}$

that

$$\begin{array}{ccccc}
\mathcal{C}^o \text{Pr}(\mathcal{F}) & \longleftarrow & \mathbb{S}' & \longrightarrow & \mathcal{C}^o \text{Pr}(\mathcal{F}) \\
\epsilon_{\mathcal{F}} \downarrow & & \downarrow k & & \downarrow \epsilon_{\mathcal{F}} \\
\mathcal{F} & \xleftarrow{L} & \mathbb{S} & \xrightarrow{R} & \mathcal{F}
\end{array}$$

commutes, for a some (unique) map k . Here \mathbb{S}' is the isomorphism family of $\text{Pr}(\mathbb{S})$, so consists of all those subsets

$$\{([L(s)]_{L\sigma}, [R(s)]_{R\sigma}) \mid s \in \sigma\}$$

for $\sigma \in \mathbb{S}$; its symmetry maps are given by the left and right projections. By assumption $\langle L, R \rangle$ is monic in \mathcal{Fam} , and so injective as a function. Because of this we obtain a well-defined function k by specifying that it takes the pair of primes $([L(s)]_{L\sigma}, [R(s)]_{R\sigma})$ to s . If $\theta \in \mathbb{S}$, then $\theta = \{([L(s)]_{L\sigma}, [R(s)]_{R\sigma}) \mid s \in \sigma\}$, for some $\sigma \in \mathbb{S}$. Clearly $k\theta = \sigma$ and k is easily seen to be locally injective. Hence we obtain $k : \mathbb{S}' \rightarrow \mathbb{S}$, a map in \mathcal{Fam} , which is readily observed to make the diagram commute.

The adjunction $\mathcal{C}^o \dashv \text{Pr}$ from \mathcal{E} to \mathcal{Fam} gives a bijection, with the following mutual inverses,

$$\mathcal{E}(A, \text{Pr}(\mathcal{F})) \begin{array}{c} \xrightarrow{\epsilon_{\mathcal{F}} \circ \mathcal{C}^o(-)} \\ \xleftarrow{\text{Pr}(-) \circ \eta_A} \end{array} \mathcal{Fam}(\mathcal{C}^o(A), \mathcal{F}) ,$$

for $A \in \mathcal{E}$ and $\mathcal{F} \in \mathcal{Fam}$. As on maps I coincides with \mathcal{C}^o and \mathcal{SPr} with Pr , we obtain the bijection of the adjunction $I \dashv \mathcal{SPr}$,

$$\mathcal{SE}(A, \mathcal{SPr}(\mathcal{F})) \begin{array}{c} \xrightarrow{\epsilon_{\mathcal{F}} \circ I(-)} \\ \xleftarrow{\mathcal{SPr}(-) \circ \eta_A} \end{array} \mathcal{SFam}(I(A), \mathcal{F}) ,$$

when $A \in \mathcal{SE}$ and $\mathcal{F} \in \mathcal{SFam}$. Because I , \mathcal{SPr} and composition preserve \sim , the bijection preserves and reflects \sim . \square

The categories \mathcal{SE} and \mathcal{SFam} are enriched in the category of equivalence relations; accordingly the adjunction is enriched—the natural bijection of the adjunction is an isomorphism of equivalence relations. Regarding \mathcal{SE} and \mathcal{SFam} as 2-categories, the adjunction is an adjunction of 2-categories.

6 Constructions in \mathcal{SE}

We first examine products in \mathcal{SE} and the meaning of their symmetry in terms of their isomorphism families.

Theorem 6.1 *Let (A, l_A, r_A) and (B, l_B, r_B) be event structures with symmetry. Their product in \mathcal{SE} is given by $(A \times B, l_A \times l_B, r_A \times r_B)$, based on the product $A \times B$ of their underlying event structures in \mathcal{E} , and sharing the same projections, $\pi_1 : A \times B \rightarrow A$ and $\pi_2 : A \times B \rightarrow B$.*

The isomorphism family of the product consists of all order isomorphisms $\theta : x \cong x'$ between finite configurations x, x' of $A \times B$, with order inherited from the

product, for which $\theta_A = \{(\pi_1(p), \pi_1(p')) \mid ((p, p') \in \theta)\}$ is in the isomorphism family of A and $\theta_B = \{(\pi_2(p), \pi_2(p')) \mid ((p, p') \in \theta)\}$ is in the isomorphism family of B .

Let $f, f' : C \rightarrow A$ and $g, g' : C \rightarrow B$ in \mathcal{E} . If $f \sim f'$ and $g \sim g'$, then $\langle f, g \rangle \sim \langle f', g' \rangle$.

Proof. Consider event structures, A with symmetry $l_A, r_A : S_A \rightarrow A$, and B with symmetry $l_B, r_B : S_B \rightarrow B$. Their product in \mathcal{E} is built from the products $A \times B$, π_1, π_2 and $S_A \times S_B, \Pi_1, \Pi_2$ in \mathcal{E} . It is routine to check that the product in \mathcal{E} is given by $A \times B$ with symmetry $l_A \times l_B, r_A \times r_B : S_A \times S_B \rightarrow A \times B$, with the same projections π_1 and π_2 as the underlying event structure. Note that the product of open maps is open. That $\langle l_A \times l_B, r_A \times r_B \rangle$ forms an equivalence relation follows point for point from $\langle l_A, r_A \rangle$ and $\langle l_B, r_B \rangle$ forming equivalence relations.

Write $\mathbb{S}_{A \times B}, \mathbb{S}_A$ and \mathbb{S}_B for the isomorphism families of $A \times B, A$ and B in \mathcal{E} , respectively. We now show

$$\theta \in \mathbb{S}_{A \times B} \iff \theta \in \mathcal{C}^o(A \times B) \times \mathcal{C}^o(A \times B) \ \& \ \theta_A \in \mathbb{S}_A \ \& \ \theta_B \in \mathbb{S}_B . \quad (\dagger)$$

“ \Rightarrow ”: Let $\theta \in \mathbb{S}_{A \times B}$. Through belonging to an isomorphism family, θ is automatically in the product of stable families $\mathcal{C}^o(A \times B) \times \mathcal{C}^o(A \times B)$. By definition,

$$\theta = \{(l_A \times l_B(s), r_A \times r_B(s)) \mid s \in z\}$$

for some $z \in \mathcal{C}^o(S_A \times S_B)$. Hence,

$$\begin{aligned} \theta_A &= \{(\pi_1(p), \pi_1(p')) \mid ((p, p') \in \theta)\} \\ &= \{(\pi_1(l_A \times l_B)(s), \pi_2(r_A \times r_B)(s)) \mid s \in z\} \\ &= \{(l_A \Pi_1(s), r_A \Pi_2(s)) \mid s \in z\} \\ &= \{(l_A(s_1), r_A(s_1)) \mid s_1 \in \Pi_1 z\} . \end{aligned}$$

As $\Pi_1 z \in \mathcal{C}^o(A)$ we obtain $\theta_A \in \mathbb{S}_A$. Similarly, $\theta_B \in \mathbb{S}_B$.

“ \Leftarrow ”: To show the converse we use a more convenient description of the product of event structures with symmetry, obtained from the coreflection of Theorem 5.2 between event structures with symmetry and stable families with symmetry:

$$\mathcal{E} \begin{array}{c} \xrightarrow{I} \\ \perp \\ \xleftarrow{SPr} \end{array} \mathcal{SFam}$$

Under the left adjoint I the event structures with symmetry (A, l_A, r_A) and (B, l_B, r_B) are sent to stable families with symmetry, to respectively, the isomorphism family \mathbb{S}_A , with maps $(a, a') \mapsto a$ and $(a, a') \mapsto a'$, and the isomorphism family \mathbb{S}_B , with maps $(b, b') \mapsto b$ and $(b, b') \mapsto b'$. Their product in \mathcal{SFam} is constructed out of the product of stable families $\mathbb{S}_A \times \mathbb{S}_B$ with projection maps $(a, a', b, b') \mapsto (a, b)$ and $(a, a', b, b') \mapsto (a', b')$ to $\mathcal{C}^o(A) \times \mathcal{C}^o(B)$. Right adjoints preserve products, so under SPr we obtain what will be a more convenient description of the product of

(A, l_A, r_A) and (B, l_B, r_B) as

$$\begin{array}{ccc} & \text{Pr}(\mathbb{S}_A \times \mathbb{S}_B) & \\ & \swarrow L & \searrow R \\ A \times B & & A \times B \end{array}$$

where L and R act on primes as follows: for $\zeta \in \mathbb{S}_A \times \mathbb{S}_B$ and $(a, a', b, b') \in \zeta$,

$$L([(a, a', b, b')]_{\zeta}) = [(a, b)]_{\zeta_1} \text{ and } R([(a, a', b, b')]_{\zeta}) = [(a', b')]_{\zeta_2}$$

where $\zeta_1 = \{(a, b) \mid (a, a', b, b') \in \zeta\}$ and $\zeta_2 = \{(a', b') \mid (a, a', b, b') \in \zeta\}$.

With the above description of the product of (A, l_A, r_A) and (B, l_B, r_B) , we obtain that the isomorphism family of the product $\mathbb{S}_{A \times B}$ consists of all sets

$$\theta = \{(L(s), R(s)) \mid s \in z\}$$

for some $z \in \mathcal{C}^o(\text{Pr}(\mathbb{S}_A \times \mathbb{S}_B))$. But configurations in $\mathcal{C}^o(\text{Pr}(\mathbb{S}_A \times \mathbb{S}_B))$ are precisely those subsets z for which

$$z = \{[(a, a', b, b')]_{\zeta} \mid (a, a', b, b') \in \zeta\}$$

for some $\zeta \in \mathbb{S}_A \times \mathbb{S}_B$. It follows that $\mathbb{S}_{A \times B}$ consists of all

$$\theta = \{([(a, b)]_{\zeta_1}, [(a', b')]_{\zeta_2}) \mid (a, a', b, b') \in \zeta\}$$

for some $\zeta \in \mathbb{S}_A \times \mathbb{S}_B$.

Now suppose $\theta \in \mathcal{C}^o(A \times B) \times \mathcal{C}^o(A \times B)$ such that $\theta_A \in \mathbb{S}_A$ and $\theta_B \in \mathbb{S}_B$. Defining

$$\zeta = \{(a, a', b, b') \mid \exists (p, p') \in \theta. \pi_1(p) = a \ \& \ \pi_1(p') = a' \ \& \ \pi_2(p) = b \ \& \ \pi_2(p') = b'\}$$

we obtain $\zeta \in \mathbb{S}_A \times \mathbb{S}_B$ for which

$$\theta = \{([(a, b)]_{\zeta_1}, [(a', b')]_{\zeta_2}) \mid (a, a', b, b') \in \zeta\},$$

as required for $\theta \in \mathbb{S}_{A \times B}$.

We have established (†). It follows that

$$\theta \in \mathbb{S}_{A \times B} \iff \theta \text{ is an order isomorphism \& } \theta_A \in \mathbb{S}_A \ \& \ \theta_B \in \mathbb{S}_B .$$

[By ‘ θ is an order isomorphism’ is meant that θ is an order isomorphism $\theta : x \cong x'$ between finite configurations $x = \{p \mid \exists p'. (p, p') \in \theta\}$ and $x' = \{p' \mid \exists p. (p, p') \in \theta\}$ of $A \times B$.] “ \Rightarrow ”: If $\theta \in \mathbb{S}_{A \times B}$, then by property (ii) of isomorphism families, θ is an order isomorphism. “ \Leftarrow ”: If θ is an order isomorphism $\theta : x \cong x'$ between finite configurations x and x' of $A \times B$, then certainly $\theta \in \mathcal{C}^o(A \times B) \times \mathcal{C}^o(A \times B)$ from

the description of the product of stable families in Appendix.

Suppose $f \sim f'$ and $g \sim g'$, where $f, f' : C \rightarrow A$ and $g, g' : C \rightarrow B$ in \mathcal{SE} . Let $y \in \mathcal{C}^o(C)$. The bijection

$$\theta_y : \langle f, g \rangle y \cong \langle f', g' \rangle y$$

is in $\mathcal{C}^o(A \times B) \times \mathcal{C}^o(A \times B)$. Because $f \sim f'$ the bijection $(\theta_y)_A : f y \cong f' y$ is in \mathbb{S}_A , and similarly $(\theta_y)_B : g y \cong g' y$ is in \mathbb{S}_B . Hence, by the above, $\theta_y \in \mathbb{S}_{A \times B}$. Thus $\langle f, g \rangle \sim \langle f', g' \rangle$. \square

The category \mathcal{SE} does not have a terminal object. However, the event structure with symmetry ϖ defined in Example 3.6 satisfies an appropriately weakened property (it is a simple instance of a biterminal object):

Proposition 6.2 *For any event structure with symmetry A there is a map $f : A \rightarrow \varpi$ in \mathcal{SE} and moreover for any two maps $f, g : A \rightarrow \varpi$ we have $f \sim g$.*

Proof. Let A be an event structure with symmetry. Because A is countable, there is clearly a map from A to ϖ . Assume two maps $f, g : A \rightarrow \varpi$. Let $x \in \mathcal{C}^o(A)$. Then $f x$ and $g x$ are sets of the same size as x . Hence $\theta_x : f x \cong g x$, where $\theta_x = \{(f(e), g(e)) \mid e \in x\}$, is in the isomorphism family of ϖ . \square

The category \mathcal{SE} does not have pullbacks and equalizers in general. However:

Proposition 6.3

(i) *Let $f, g : A \rightarrow B$ be two maps between event structures with symmetry. They have a pseudo equalizer, i.e. an event structure with symmetry E and map $e : E \rightarrow A$ such that $f \circ e \sim g \circ e$ which satisfies the further property that for any event structure with symmetry E' and map $e' : E' \rightarrow A$ such that $f \circ e' \sim g \circ e'$, there is a unique map $h : E' \rightarrow E$ such that $e' = e \circ h$.*

(ii) *Let $f : A \rightarrow C$ and $g : B \rightarrow C$ be two maps between event structures with symmetry. They have a pseudo pullback, i.e. an event structure with symmetry D and maps $p : D \rightarrow A$ and $q : D \rightarrow B$ such that $f \circ p \sim g \circ q$ which satisfies the further property that for any event structure with symmetry D' and maps $p' : D' \rightarrow A$ and $q' : D' \rightarrow B$ such that $f \circ p' \sim g \circ q'$, there is a unique map $h : D' \rightarrow D$ such that $p' = p \circ h$ and $q' = q \circ h$.*

Proof. (i) Let \mathbb{S}_B be the isomorphism family of B . Define a family of finite configurations

$$\mathcal{D} = \{x \in \mathcal{C}^o(A) \mid \phi_x : f x \cong g x \text{ in } \mathbb{S}_B\}$$

—we use ϕ_x for the bijection between $f x$ and $g x$ induced by the local injectivity of f and g w.r.t. a finite configuration x . Define the event structure E to comprise events $\{a \in A \mid \exists x \in \mathcal{D}. a \in x\}$ with causal dependency the restriction of that in A and consistency, $X \in \text{Con}_D$ iff $\exists x \in \mathcal{D}. X \subseteq x$. Observe that

$$\forall x' \in \mathcal{C}^o(A). x' \subseteq x \in \mathcal{D} \Rightarrow x' \in \mathcal{D}$$

because any bijection $\phi_x : f x \cong g x$ in \mathbb{S}_B restricts to a bijection $\phi_{x'} : f x' \cong g x'$ also in \mathbb{S}_B . It follows that \mathcal{D} coincides with the family of finite configurations $\mathcal{C}^o(E)$.

We tentatively define an isomorphism family \mathbb{S}_E on E as follows. For $x, y \in \mathcal{C}^o(E)$, take $\theta : x \cong y$ to be in \mathbb{S}_E iff $\theta : x \cong y$ is in \mathbb{S}_A . We need to verify that \mathbb{S}_E is indeed an isomorphism family, for which we require properties (i), (ii), (iii) of Definition 3.2. Properties (i) and (ii) follow directly from the corresponding properties of \mathbb{S}_A , where the observation above is used for (ii).

To show (iii), assume $\theta : x \cong y$ is in \mathbb{S}_E and $x \subseteq x' \in \mathcal{C}^o(E)$. Then

$$\phi_x : fx \cong gx \text{ and } \phi_{x'} : fx' \cong gx' \text{ are in } \mathbb{S}_B .$$

As $\theta : x \cong y$ is also in \mathbb{S}_A , there is an extension $\theta' : x' \cong y'$ in \mathbb{S}_A , for some $y' \in \mathcal{C}^o(A)$. We require that $y' \in \mathcal{C}^o(E)$, for which it suffices to show $y' \in \mathcal{D}$. However, as f and g preserve symmetry,

$$f\theta' : fx' \cong fy' \text{ and } g\theta' : gx' \cong gy' \text{ are in } \mathbb{S}_B .$$

Hence $\phi_{y'} = (g\theta') \circ \phi_{x'} \circ (f\theta')^{-1} : fy' \cong gy'$ is in \mathbb{S}_B . So $y' \in \mathcal{D}$, as required.

(ii) This now follows as we can construct pseudo pullbacks from pseudo equalizers and products: two maps $f : A \rightarrow C$ and $g : B \rightarrow C$ between event structures with symmetry have a pseudo pullback given as the pseudo equalizer of the two maps $f\pi_1, g\pi_2 : A \times B \rightarrow C$, got by composing with projections of the product. \square

There are obvious weakenings of the conditions of (i) and (ii) in which the uniqueness is replaced by uniqueness up to \sim and equality by \sim —these are simple special cases of bilimits called biequalizers and bipullbacks when we regard \mathcal{E} as a 2-category. As in the Proposition 6.3, we follow tradition and call the stricter construction described in (ii) a *pseudo pullback*. In Theorem 6.1, that pairing of maps preserves \sim means that the products described are 2-products in \mathcal{E} regarded as a 2-category. An accessible introduction to limits in 2-categories is [18].

7 Functors and pseudo monads

By Proposition 5.1, certain functors on the category of event structures \mathcal{E} , straightforwardly induce functors on \mathcal{S} , the enriched category of event structures with symmetry. A functor on several, even infinitely many, arguments $F : \mathcal{E} \times \cdots \times \mathcal{E} \times \cdots \rightarrow \mathcal{E}$ which preserves pullbacks, open maps and has monic mediators for products will induce a functor on event structures with symmetry respecting \sim on homsets. (A map in a product of categories, such as $\mathcal{E} \times \cdots \times \mathcal{E} \times \cdots$, is taken to be open iff it is open in each component.) We consider some examples.

7.1 Operations

7.1.1 Simple parallel composition

For example, consider the functor $\parallel : \mathcal{E} \times \mathcal{E} \rightarrow \mathcal{E}$ which given two event structures puts them in parallel. Let $(A, \text{Con}_A, \leq_A)$ and $(B, \text{Con}_B, \leq_B)$ be event structures. The events of $A \parallel B$ are $(\{0\} \times A) \cup (\{1\} \times B)$; with $(0, a) \leq (0, a')$ iff $a \leq_A$

a' and $(1, b) \leq (1, b')$ iff $b \leq_B b'$; and with a subset of events C consistent in $A \parallel B$ iff $\{a \mid (0, a) \in C\} \in \text{Con}_A$ and $\{b \mid (1, b) \in C\} \in \text{Con}_B$. The operation extends to a functor—put the two maps in parallel. It is not hard to check that the functor \parallel preserves pullbacks and open maps, and that the mediating maps $(A \times A) \parallel (B \times B) \rightarrow (A \parallel B) \times (A \parallel B)$ are monic. Consequently it induces a functor $\parallel: \mathcal{SE} \times \mathcal{SE} \rightarrow \mathcal{SE}$ which preserves \sim on homsets. On the same lines the functor giving the parallel composition $\parallel_{i \in I} A_i$ of countably-indexed event structures $A_i, i \in I$, extends to a functor on event structures with symmetry.

7.1.2 Sum

Similarly, the coproduct or sum of two event structures extends to the sum of event structures with symmetry. Let $(A, \text{Con}_A, \leq_A)$ and $(B, \text{Con}_B, \leq_B)$ be event structures. The events of the sum $A+B$ are $(\{0\} \times A) \cup (\{1\} \times B)$; with $(0, a) \leq (0, a')$ iff $a \leq_A a'$ and $(1, b) \leq (1, b')$ iff $b \leq_B b'$; but now a subset of events C is consistent in $A+B$ iff there is $C_0 \in \text{Con}_A$ such that $C = \{(0, a) \mid a \in C_0\}$ or there is $C_1 \in \text{Con}_B$ such that $C = \{(1, a) \mid a \in C_1\}$. We can also form a sum $\Sigma_{i \in I} A_i$ of event structures A_i indexed by a countable set I . Again this extends to a functor on event structures with symmetry.

7.2 An enriched adjunction

Adding symmetry, following the general procedure of Section 5, starting from the category of event structures with rigid maps \mathcal{E}_r , we form \mathcal{SE}_r ; we end up with exactly the same objects, event structures with symmetry, but with rigid maps, preserving symmetry, between them. The adjunction of Proposition 2.3, relating rigid and total maps, lifts to an adjunction, enriched in equivalence relations,

$$\mathcal{SE}_r \begin{array}{c} \xleftarrow{\text{Saug}} \\ \xrightarrow{\top} \end{array} \mathcal{SE}$$

between event structures with symmetry.

This is essentially because the inclusion functor and its right adjoint aug lift to their counterparts with symmetry. The inclusion functor $\mathcal{E}_r \hookrightarrow \mathcal{E}$ meets the conditions of Proposition 5.1 yielding an inclusion functor $\mathcal{SE}_r \hookrightarrow \mathcal{SE}$. The open maps are the same in the two categories; products w.r.t. rigid maps are got by restricting those w.r.t. total maps, ensuring monic mediators for products; and pullbacks w.r.t. rigid maps are necessarily also pullbacks w.r.t. total maps. The right adjoint $\text{aug} : \mathcal{E} \rightarrow \mathcal{E}_r$ automatically preserves pullbacks and products, making the mediating maps for products isomorphisms so certainly monic, and again open maps coincide in the two categories. So it too lifts, via Proposition 5.1, to $\text{Saug} : \mathcal{SE} \rightarrow \mathcal{SE}_r$.

Naturality of the unit η and counit of the original adjunction between \mathcal{E}_r and

\mathcal{E} , ensures that their components preserve symmetry. We thus obtain a bijection

$$\mathcal{SE}_r(A, \text{aug}(B)) \begin{array}{c} \xrightarrow{\epsilon_B \circ (-)} \\ \xleftarrow{\text{Saug}(-) \circ \eta_A} \end{array} \mathcal{SE}(A, B) ,$$

for event structures with symmetry A and B . Because Saug and composition preserve \sim , the bijection preserves and reflects \sim . Viewing the categories with symmetry as enriched in the category of equivalence relations, the adjunction is enriched. Viewing the categories as 2-categories, the adjunction is a 2-adjunction.

Correspondingly, the monad aug on \mathcal{E}_r lifts to an enriched monad, and 2-monad, Saug on \mathcal{SE}_r . Its Kleisli category is isomorphic to \mathcal{SE} . (We shall soon see that \mathcal{SE}_p , the category of event structures with symmetry based on partial maps, can be obtained as a Kleisli construction from a pseudo monad on \mathcal{SE} , so also from a pseudo monad on \mathcal{SE}_r .)

It is instructive to work out the biterminal object \top in \mathcal{SE}_r (recalling Example 3.6 and Proposition 2.3). It is obtained from the biterminal object ϖ of \mathcal{SE} as its image under the right adjoint Saug . The finite configurations of \top correspond to partial orders on finite subsets of natural numbers ω ; inclusion on configurations corresponds to the rigid order between partial orders, described in the proof of Proposition 2.3. Viewing \top 's configurations in this way, its isomorphism family consists of partial-order isomorphisms between partial orders on finite subsets of ω .

7.3 Pseudo monads

That categories with symmetry are enriched over equivalence relations ensures that they support the definitions of 2-functor, 2-natural transformation, 2-adjunction and 2-monad which respect \sim ; in this simple case 2-natural transformations coincide with natural transformations. Categories with symmetry also support the definitions of pseudo functor and pseudo natural transformation, which parallel those of functor and natural transformation, but with equality replaced by \sim . In the same spirit a pseudo monad satisfies variants of the usual monad laws but expressed in terms of \sim rather than equality (we can ignore the extra coherence conditions [5] as they trivialize in the simple situation here). As examples we consider two particular pseudo monads which we can apply to the semantics of higher-order nondeterministic processes. (There is an attendant weakening of the notions of adjunction and equivalence between categories to that of biadjunction and biequivalence.)

The following examples are based on constructions we have seen earlier.

7.3.1 The copying pseudo monad

The copying operation $!$ of Example 3.7 extends to a functor on \mathcal{SE} . Let $f : A \rightarrow B$ be a map of event structures with symmetry. Define $!f : !A \rightarrow !B$ by taking $!f(i, a) = (i, f(a))$ for all events a of A . The functor $!$ preserves \sim on homsets. (It is not induced by a functor on \mathcal{E} .)

The component of the unit $\eta_E^! : E \rightarrow !E$ acts so $\eta_E^!(e) = (0, e)$ for all events $e \in E$ —it takes an event structure with symmetry E into its zeroth copy in $!E$.

The multiplication map relies on a subsidiary pairing function on natural numbers $[-, -] : \omega \times \omega \rightarrow \omega$ which we assume is injective. The component of the multiplication $\mu_E^! : !!E \rightarrow !E$ acts so $\mu_E^!(i, j, e) = ([i, j], e)$.

It can be checked that the unit and the multiplication are natural transformations and that the usual monad laws, while they do not hold up to equality, do hold up to \sim . The somewhat arbitrary choice of the zeroth copy in the definition of the unit and pairing function on natural numbers in the definition of the multiplication don't really matter in the sense that other choices would lead to components \sim -equivalent to those chosen. (Different choices lead to natural transformations related by modifications with \sim at all components.)

7.3.2 The partiality pseudo monad

Let E be an event structure with symmetry. Define $E_* =_{\text{def}} E \parallel \varpi$, i.e. it consists of E and ϖ put in parallel.

The component of the unit $\eta_E^* : E \rightarrow E_*$ acts so $\eta_E^*(e) = (0, e)$ for all events $e \in E$ —so taking E to its copy in $E \parallel \varpi$.

The component of the multiplication $\mu_E^* : (E_*)_* \rightarrow E_*$ acts so $\mu_E^*(0, (0, e)) = (0, e)$ and $\mu_E^*(0, (1, j)) = [0, j]$ and $\mu_E^*(1, k) = [1, k]$, where we use the pairing function on natural numbers above to map the two disjoint copies of ω injectively into ω .

Both η^* and μ^* are natural transformations and the usual monad laws hold up to \sim making a pseudo monad. Again, the definition of multiplication is robust; if we used some alternative way to inject $\omega + \omega$ into ω the resulting multiplication would be \sim -related at each component to the one we have defined.

The category of event structures with partial maps has played a central role in the event structure semantics of synchronizing processes [23]. It readily generalizes to accommodate symmetry. By following the general procedure of Section 5, we obtain \mathcal{SE}_p ; it has event structures with symmetry as objects but now with partial maps between them. Through exploiting symmetry, \mathcal{SE}_p now reappears as a Kleisli construction based on the pseudo monad $(-)_*$. Here we must face a technicality. Because $(-)_*$ is a pseudo monad when we follow the obvious analogue of the Kleisli construction for a monad on a category we find that the associativity and identity laws for composition only hold up to symmetry, \sim . Technically the Kleisli construction yields a very simple instance of a bicategory, where the coherence conditions trivialize. The Kleisli construction is a simple special case of that of the Kleisli bicategory of a pseudo monad described in [5].

Proposition 7.1 *The Kleisli bicategory of the pseudo monad $(-)_*$ and the category \mathcal{SE}_p of event structures with symmetry and partial maps (regarded as a 2 category) are biequivalent; the biequivalence is the identity on objects and takes maps $f : A \rightarrow B_*$ in the Kleisli bicategory to partial maps $\bar{f} : A \rightarrow B$, undefined precisely when the image is in ϖ .*

Proof. To be a biequivalence we need that $f \mapsto \bar{f}$ from maps in the Kleisli bicategory $Kl((-)_*)(A, B) = \mathcal{ES}(A, B_*)$ to $\mathcal{ES}_p(A, B)$ is (essentially) onto, preserves and reflects \sim —as is easily checked. \square

7.4 Equivalences

We have enough operations to derive some useful equivalences. Below we use 1 to denote the single-event event structure with symmetry and \otimes for the product of event structures with symmetry with partial maps.

Proposition 7.2 *For event structures with symmetry:*

- (i) $!A \parallel !B \simeq !(A + B)$ and $\|_{k \in K} !A_k \simeq !\Sigma_{k \in K} A_k$ where K is a countable set.
- (ii) $\varpi \simeq !1$ and $A \times \varpi \simeq A$.
- (iii) $A_* \simeq A \parallel \varpi$, $(!A)_* \simeq !(A + 1)$ and $(A \otimes B)_* \simeq A_* \times B_*$.

Proof. (i) Define $f : !A \parallel !B \rightarrow !(A + B)$ and $g : !(A + B) \rightarrow !A \parallel !B$ as follows:

$$\begin{aligned} f(0, (i, a)) &= (2i, (0, a)); & f(1, (j, b)) &= (2j + 1, (1, b)); & \text{and} \\ g(i, (0, a)) &= (0, (i, a)); & g(j, (1, b)) &= (1, (j, b)) . \end{aligned}$$

To show $\text{id}_{!A \parallel !B} \sim gf$, consider $x \in \mathcal{C}^o(!A \parallel !B)$ for which

$$gf x = \{(0, (2i, a)) \mid (0, (i, a)) \in x\} \cup \{(1, (2j + 1, b)) \mid (1, (j, b)) \in x\} .$$

The bijection induced by gf , viz. $(0, (i, a)) \mapsto (0, (2i, a)); (1, (j, b)) \mapsto (1, (2j + 1, b))$ from x to $gf x$, is in the isomorphism family of $!A \parallel !B$. Hence $\text{id}_{!A \parallel !B} \sim gf$. Similarly, for $y \in \mathcal{C}^o(!(A + B))$,

$$fgy = \{(2i, (0, a)) \mid (i, (0, a)) \in y\} \cup \{(2j + 1, (1, b)) \mid (j, (1, b)) \in y\} .$$

This time the bijection $(i, (0, a)) \mapsto (2i, (0, a)); (j, (1, b)) \mapsto (2j + 1, (1, b))$ from y to fgy is in the isomorphism family of $!(A + B)$. Hence $\text{id}_{!(A + B)} \sim fg$.

The proof for the infinitary version, $\|_{k \in K} !A_k \simeq !\Sigma_{k \in K} A_k$, where K is a countable set, is similar. Assume an injective pairing function $[_, _] : K \times \omega \rightarrow \omega$. Define $f : \|_{k \in K} !A_k \rightarrow !\Sigma_{k \in K} A_k$ and $g : !\Sigma_{k \in K} A_k \rightarrow \|_{k \in K} !A_k$ by $f(k, (i, a)) = ([k, i], (k, a))$ and $g(i, (k, a)) = (k, (i, a))$. For $x \in \mathcal{C}^o(!\Sigma_{k \in K} A_k)$ and $y \in \mathcal{C}^o(\|_{k \in K} !A_k)$,

$$\begin{aligned} gf x &= \{(k, ([k, i], a)) \mid (k, (i, a)) \in x\} \text{ and} \\ fg y &= \{([k, i], (k, a)) \mid (i, (k, a)) \in y\} . \end{aligned}$$

The bijections induced by gf from x to $gf x$ and fg from y to $fg y$ are in the isomorphism families of $\|_{k \in K} !A_k$ and $\Sigma_{k \in K} A_k$, respectively.

(ii) $A \times \varpi \simeq A$ because both sides are biproducts of A and ϖ , so equivalent.

(iii) From Proposition 7.1, there is a biadjunction $\mathcal{SE} \xleftarrow{(-)_*} \mathcal{SE}_p$. The right biadjoint $(-)_*$ preserves products up to equivalence, so $(A \otimes B)_* \simeq A_* \times B_*$.

The remaining equivalences are obvious. \square

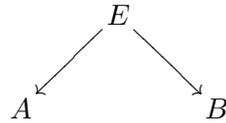
The equivalence $!A \parallel !B \simeq !(A + B)$, and its infinite version in (i), express the sense in which copying obviates choice. More importantly, they and the other the equivalences enable definitions by case analysis on events, also in the presence of asynchrony.

8 Applications

We briefly sketch some applications, the subject of present and future work, and less finished than that of the previous sections.

8.1 Spans

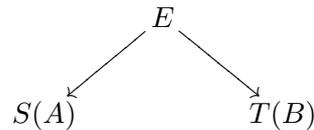
Because \mathcal{E} has pseudo pullbacks—Proposition 6.3, we can imitate the standard construction of the bicategory of spans to produce a bicategory $\text{Span}_{\mathcal{E}}$. Its objects are event structures with symmetry. Its maps $\text{Span}_{\mathcal{E}}(A, B)$, from A to B , are spans



composed using the pseudo pullbacks of of Proposition 6.3 (ii). Its 2-cells, maps in $\text{Span}_{\mathcal{E}}(A, B)$, are the maps between the vertices of two spans making the obvious triangles commute. $\text{Span}_{\mathcal{E}}$ has a tensor and function space given by the product of \mathcal{E} .

An individual span can be thought of as a process computing from input of type A to output of type B . But given the nature of maps in \mathcal{E} such a process is rather restricted; from a computational view the process is unnaturally symmetric and ‘ultra-linear’ because any output event is synchronized with an event of input.

We wish to modify the maps of a span to allow for different regimes of input and output. A systematic way to do this is through the use of pseudo monads on \mathcal{E} and build more general spans



for pseudo monads S and T . For example a span in which $S = (-)_*$ and $T = !(-)$ would permit output while ignoring input and allow the output of arbitrarily many similar events of type B . But for such general spans to compose, we require that S and T satisfy several conditions, which we only indicate here:

- in order to lift to pseudo comonads and monads on spans, S and T should be ‘cartesian’ pseudo monads, now w.r.t. pseudo/bipullbacks (adapting [3]);

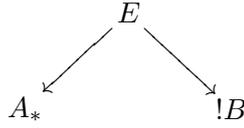
- in order to obtain a comonad-monad distributive law for the liftings of S and T to spans it suffices to have a ‘cartesian’ distributive law for S and T , with commutativity up to \sim , with extra pseudo/bipullback conditions on two of the four diagrams (adapting [13]).

The two pseudo monads $S = (-)_*$ and $T = !(-)$ do satisfy these requirements with a distributive law with components $\lambda_E : (!E)_* \rightarrow !(E_*)$ such that $\lambda_E(0, (j, e)) = (j, (0, e))$ and $\lambda_E(1, k) = (0, (1, k))$.

The paper has concentrated on the categories of event structures \mathcal{E} and \mathcal{SE} with total maps. In particular, general spans have been described for maps in \mathcal{SE} . Analogous definitions and results hold for rigid maps, and for spans in \mathcal{SE}_r . Recall from Section 7.2 that total maps on event structures with symmetry can be obtained as Kleisli maps w.r.t. a monad \mathcal{Saug} on \mathcal{SE}_r . It appears that we can ground all the maps and spans of event structures of interest in \mathcal{SE}_r . The category \mathcal{SE}_r is emerging as *the* fundamental category of event structures.

8.2 Event types

The particular bicategory of spans



is already quite an interesting framework for the semantics of higher-order processes. It supports types including:

- Prefix types $!T$: in which a single event \bullet prefixes T for an event structure with symmetry T .
- Sum types $\Sigma_{\alpha \in A} T_\alpha$: the sum of a collection T_α , for $\alpha \in A$, of event structures with symmetry—the sum functor is described in Section 7.1.2. Sum types may also be written $a_1 T_1 + \dots + a_n T_n$ when the indexing set is finite. The empty sum type is the empty event structure \emptyset .
- Tensor types $T_1 \otimes T_2$: the product in \mathcal{SE}_p .
- Function types $T_1 \multimap T_2$: a form of function space, defined as the product $(T_1)_* \times !T_2$ in \mathcal{SE} .⁵
- Recursively defined types: treated for example as in [23,25].

The types describe the events and basic causalities of a process, and in this sense are examples of *event types*, or *causal types*, of a process. (One can imagine other kinds of spans and variations in the nature of event types.)

⁵ Although this function space seems hard to avoid for this choice of span and tensor, we don’t quite have $\multimap B$ a left biadjoint to $B \multimap \dots$

As an example, the type of a process only able to do actions within a_1, \dots, a_k could be written

$$a_1 \bullet !\emptyset + \dots + a_k \bullet !\emptyset,$$

which we condense to $a_1 + \dots + a_k$, as it comprises the event structure with events a_1, \dots, a_k made in pairwise-conflict, with the identity relation of causal dependency. The judgement that a closed process, represented by an event structure with symmetry E , has this type would be associated with a degenerate span from the biterminal \emptyset_* to $!(a_1 + \dots + a_k)$, so essentially with a map

$$l : E \rightarrow !(a_1 + \dots + a_k)$$

in \mathcal{SE} , ‘labelling’ events by their actions. By Proposition 7.2 (i), there is an equivalence

$$!a_1 \parallel \dots \parallel !a_k \simeq !(a_1 + \dots + a_k),$$

and a process of this type can only do actions a_1, \dots, a_k , though with no bound on how many times any action can be done.

The type of CCS, with channels A , can be written as

$$Act = \tau \bullet !\emptyset + \sum_{\bar{a} \in \bar{A}} \bar{a} \bullet !\emptyset + \sum_{a \in A} a \bullet !\emptyset.$$

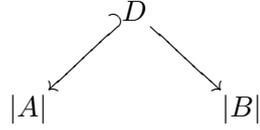
We can describe the parallel composition of CCS by a partial function from the events $Act \otimes Act$ to the events $!Act$, expressing how events combine to form synchronization events (the second line), or can occur asynchronously (the first):

$$\begin{aligned} (\alpha, *) &\mapsto \mu_{Act}^!(0, \eta_{Act}^!(\alpha)), & (*, \alpha) &\mapsto \mu_{Act}^!(1, \eta_{Act}^!(\alpha)), \\ (a, \bar{a}), (\bar{a}, a) &\mapsto \eta_{Act}^!(\tau), & &\text{and undefined otherwise.} \end{aligned}$$

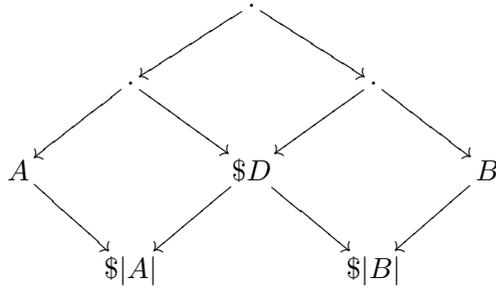
This partial function is also a partial map of event structures from $Act \otimes Act$ to $!Act$ —it would have violated local injectivity and not been a map of event structures, had we chosen simply $\eta_{Act}^!(\alpha)$ as the resulting events in the first two clauses. The partial function is readily interpreted as a span from $Act \otimes Act$ to $!Act$ —its vertex is essentially the domain of definition of the partial function. Post-composing its left ‘leg’ with $\eta_{Act \otimes Act}^*$ we obtain a span from $(Act \otimes Act)_*$ to $!Act$ which denotes the parallel composition of CCS. Given two CCS processes represented by degenerate spans, we can combine them to a process with event type $Act \otimes Act$, denoting a degenerate span ending in $!(Act \otimes Act)$. Its composition with the span for parallel composition can be shown to give the traditional event-structure semantics of parallel composition in CCS [23,25,26,21].

In fact there is a general way to define spans from partial functions on events which respect symmetry. There is a functor from event structures with symmetry \mathcal{SE} to equivalence relations; it takes an event structure with symmetry A to the equivalence relation $|A|$ induced by the symmetry on the set of events. The functor is enriched in equivalence relations and has a right biadjoint $\$$ which takes an equivalence relation $(L, R \subseteq L \times L)$ to the event structure with symmetry $!(L, l, r : R \rightarrow L)$, where we understand L as an event structure with events in pairwise conflict with

trivial causal dependency, R similarly, and with symmetry maps given as the obvious projections from R to L . (The biadjunction between event structures with symmetry and equivalence relations relies on the event structures being consistent-countable—see the footnote accompanying the definition of event structures, Section 2.) For event structures with symmetry A and B , a partial function respecting equivalence relations from $|A|$ to $|B|$ can be regarded as a span



in the category of equivalence relations—the equivalence relation D being where the partial function is defined. The unit of the biadjunction with equivalence relations has components $A \rightarrow \$|A|$ and $B \rightarrow \$|B|$, so by applying $\$$ to the span above and taking successive pseudo pullbacks we obtain a span from A to B :



A partial function between on events may not be so simple to define directly by case analysis on events. This is because the events that arise in products of event structures can be quite complicated; the events of a product $A \otimes B$ of event structures A and B are perhaps best seen as prime configurations of a product of stable families—see Appendix B. Their complexity contrasts with the simplicity of the events arising in constructions on stable families; the events of the corresponding product of stable families are simple pairs $(a, *)$, $(*, b)$ and (a, b) , where a and b are events of the components. For this reason it can be easiest to define a partial map on event structures (so a partial function on their events) via a partial map between their representations as stable families. This is so below, in a putative ‘true concurrency’ definition of a version of higher-order CCS and its parallel composition.

A form of higher-order CCS could reasonably be associated with the recursive type

$$T = \tau \bullet T + \Sigma_{\bar{a} \in \bar{A}} \bullet (T \otimes T) + \Sigma_{a \in A} \bullet (T \multimap T),$$

specifying that an event of a higher-order CCS process is either a ‘process’ event following a τ -event, a ‘concretion’ event following an output synchronization $\bar{a} \in \bar{A}$, or an ‘abstraction’ event following an input synchronization $a \in A$. Why is the first component in the type T of the form $\tau \bullet T$ and not just $\tau \bullet \emptyset$? Without the present choice I cannot see how to ensure that in the parallel composition an inter-

action between a concretion and abstraction event always follows a corresponding synchronization at their channels.

Parallel composition in higher-order CCS would be associated with a typing judgment $x : T, y : T \vdash (x | y) : T$. The typing judgment should denote a span from $(T \otimes T)_*$ to $!T$. As above, we can define a tentative parallel composition via a partial function from $|T \otimes T|$ to $!|T|$. The partial function should describe when and how events of T combine. Because events of the product $T \otimes T$ are quite complicated we must face the difficulties outlined above. However, first we need a makeshift syntax for events in T . Events of higher-order CCS are either internal events τ , subsequent process events $\tau.(i, t)$, output synchronizations \bar{a} , subsequent concretion events $\bar{a}.(i, c)$, input synchronizations a , or subsequent abstraction events $a.(j, f)$ —the natural numbers i, j index the copies in $!$ -types. In the notation for events of the product $T \otimes T$ we exploit the way it is built from a product of stable families; in the product of stable families out of which $T \otimes T$ is constructed events have the simple form of pairs $(t, *)$, $(*, t)$ or (t, t') , where t and t' are events of T . We can define a partial map from this stable family to the stable family of $!T$ by case analysis on events:

$$\begin{aligned}
t | * &= \mu_T^!(0, \eta_T^!(t)), & * | t &= \mu_T^!(1, \eta_T^!(t)), \\
a | \bar{a} &= \bar{a} | a = \eta_T^!(\tau), \\
a.(i, f) | \bar{a}.(j, c) &= \bar{a}.(i, c) | a.(j, f) = \eta_T^!(\tau \cdot \mu_T^!([i, j], (f | c))) \\
&\text{provided } (f | c) \text{ is defined,} \\
\tau.(i, t) | \tau.(j, t') &= \mu_T^!([i, j], (t | t')) \text{ provided } (t | t') \text{ is defined,} \\
\tau.(i, t) | \alpha &= \mu_T^!(i, (t | \alpha)), & \alpha | \tau.(j, t) &= \mu_T^!(j, (\alpha | t)) \\
&\text{provided } \alpha \text{ is not of the form } \tau.(k, t''), & &\text{and undefined otherwise.}
\end{aligned}$$

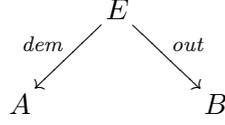
We have combined indices i, j using an injective pairing $[i, j]$ of natural numbers. The definition above relies on our simultaneously defining not just how process events combine, but also how ‘abstraction’ events f in type $T \multimap T$ and ‘concretion’ events c in type $T \otimes T$ combine to form a process event $(f | c)$ in type $!T$. We postpone the full definition. Although provisional, I hope the example helps illustrate the aims and present difficulties.

Clearly the syntax of operations to accompany the types is unfinished and really needed. But I believe the examples indicate the potential of a more thorough study of event types and give a flavour of the style of definition they might support, a method of definition which breaks away from traditional ‘interleaving’ approaches to concurrency.

8.3 Nondeterministic dataflow and affine-HOPLA

‘Stable’ spans of event structures have been used to give semantics to nondeterministic dataflow [19] and the higher-order process language affine-HOPLA [16]. They are generalisations of Berry’s stable functions [2]: deterministic stable spans

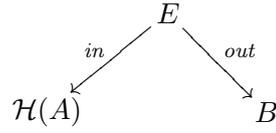
correspond to stable functions—see [19]. A stable span



consists of a ‘demand’ map $\textit{dem} : E \rightarrow A$ and a rigid map $\textit{out} : E \rightarrow B$. That \textit{dem} is a demand map means that it is a function from $\mathcal{C}^o(A)$ to $\mathcal{C}^o(B)$ which preserves unions of configurations when they exist. An equivalent way to view the demand map \textit{dem} is as a function from the events of E to finite configurations of A such that if $e \leq e'$ then $\textit{dem}(e) \subseteq \textit{dem}(e')$, and if $X \in \text{Con}$ then $\textit{dem}X \uparrow$, *i.e.*, the demands are compatible. The intuition is that $\textit{dem}(e)$ is the minimum input required for the event e to occur; when it does $\textit{out}(e)$ is observed in the output. (The stable span is *deterministic* when $\textit{dem}X \uparrow$ implies $X \in \text{Con}$, for X a finite subset of events in E .)

On the face of it demand maps are radically different from rigid maps of event structures. They can however be recovered as Kleisli maps associated with a pseudo monad \mathcal{H} on event structures with symmetry and rigid maps, as will be described shortly.

Roughly the pseudo monad \mathcal{H} adjusts the nature of events so that they record the demand history on the input. This enables stable spans to be realized as spans



of rigid maps in \mathcal{SE}_r . Such spans are a special case of the general spans of Section 8.1, with the identity monad on the right-hand-side. Because of ‘Seely conditions’ $\mathcal{H}(E \parallel F) \simeq \mathcal{H}(E) \times \mathcal{H}(F)$ and $\mathcal{H}(\emptyset) \simeq \top$ relating parallel composition \parallel and its unit, the empty event structure \emptyset , to product \times and the biterminal object \top in \mathcal{SE}_r , we obtain a description of the function space, w.r.t. parallel composition $A \parallel B$, as $A \multimap B = \mathcal{H}(A) \times B$. A very different route to the definition of function space using stable families is described in the PhD thesis [16].

We describe the pseudo monad \mathcal{H} via a biadjunction which induces it.

The biadjunction between rigid and demand maps

Let \mathcal{D} consist of objects, event structures with symmetry, and ‘demand’ maps from A to B those functions $d : \mathcal{C}^o(A) \rightarrow \mathcal{C}^o(B)$ which preserve unions when they exist and symmetry in the sense that if $x \stackrel{\theta}{\cong} x'$ is in the isomorphism family of A , then $dx \stackrel{\phi}{\cong} dx'$ is the isomorphism family of B , for some ϕ . Its maps compose as functions. The category \mathcal{D} is enriched in equivalence relations: two maps $d, d' : A \rightarrow B$ in \mathcal{D} are equivalent iff $dx \stackrel{\phi}{\cong} d'x$ is in the isomorphism family of B , for some ϕ , for any $x \in \mathcal{C}^o(B)$.

There is an obvious ‘inclusion’ functor $\mathcal{E} \hookrightarrow \mathcal{D}$: it is the identity on objects and takes a map $f : A \rightarrow B$ in \mathcal{E} to the map, also called f , from $\mathcal{C}^o(A)$ to $\mathcal{C}^o(B)$ in \mathcal{D} given by direct image under f . Somewhat surprisingly, the inclusion functor has a right biadjoint. The Kleisli bicategory of the adjunction is biequivalent to \mathcal{D} , regarded as a 2-category.

The definition of the right biadjoint makes essential use of symmetry. Let B be an event structure with symmetry. We describe a new event structure with symmetry $\mathcal{H}(B)$ in which configurations correspond to histories of demands. We first define histories and how they form a prime algebraic domain.

A *history* is a demand map $h : I \rightarrow B$ from an elementary event structure I with events lying in ω . We order two histories $h : I \rightarrow B$ and $h' : I' \rightarrow B$ by $h \sqsubseteq h'$ iff there is a rigid inclusion map $I \hookrightarrow I'$ such that

$$\begin{array}{ccc} I & \xrightarrow{h} & B \\ \uparrow & \nearrow h' & \\ I' & & \end{array}$$

commutes. Histories under \sqsubseteq form a prime algebraic domain in which the complete primes are those histories $p : J \rightarrow B$ for which J has a top element; given a history $h : I \rightarrow B$ the complete primes below it are exactly the restrictions $h \upharpoonright i$ of h to $[i]$, for $i \in I$.

We regard two histories $h : I \rightarrow B$ and $h' : I' \rightarrow B$ as similar, via ϕ, ξ , when $\phi : I \cong I'$ is an isomorphism of elementary event structures and ξ is a bijection $\xi : \bigcup_{i \in I} h(i) \cong \bigcup_{i' \in I'} h'(i')$ in the isomorphism family of B such that for all $i \in I$ its restrictions $\xi \upharpoonright h(i) \cong h'(\phi(i))$ are also in the isomorphism family.

As in Theorem B.1 of the appendix, we build the event structure of $\mathcal{H}(B)$ out of the complete-prime histories. We can describe the symmetry of $\mathcal{H}(B)$ as an isomorphism family. If x and x' are finite configurations of $\mathcal{H}(B)$, they consist of the complete primes below histories $h = \bigsqcup x : I \rightarrow B$ and $h' = \bigsqcup x' : I' \rightarrow B$, respectively. We put $\theta : x \cong x'$ in the isomorphism family of $\mathcal{H}(B)$ precisely when the histories h and h' are similar via some ϕ, ξ and

$$\theta = \{(h \upharpoonright i, h' \upharpoonright \phi(i)) \mid i \in I\} .$$

For an event structure with symmetry B define the demand map $\epsilon_B : \mathcal{H}(B) \rightarrow B$ on $x \in \mathcal{C}^o(\mathcal{H}(B))$ to be $\epsilon_B(x) = \bigcup_{i \in I} h(i)$, where $h = \bigsqcup x : I \rightarrow B$. It can be shown that the function

$$\epsilon_B \circ - : \mathcal{E}_r(A, \mathcal{H}(B)) \rightarrow \mathcal{D}(A, B)$$

is onto, and preserves and reflects \sim . We obtain a biadjunction

$$\mathcal{E}_r \overset{\mathcal{H}}{\underset{\top}{\dashv}} \mathcal{D} ,$$

one where \mathcal{D} is biequivalent to the Kleisli bicategory of the associated pseudo monad.

In fact the biadjunction can be factored through a biadjunction between \mathcal{SE}_r and the category of *event structures with persistence* and rigid maps [28]. There is an operation of quotienting an event structure by its symmetry, and this extends to a functor from \mathcal{SE}_r to event structures with persistence, a functor which has a right biadjoint. There is a further adjunction between event structures with persistence and event structures with demand maps [20]. Composed together the two biadjunctions yield the biadjunction from \mathcal{SE}_r to \mathcal{D} .

8.4 Unfoldings

Another application of symmetry is to the unfolding of Petri nets with multiple tokens, and the unfolding of higher-dimensional automata (hda's) [7]. Unfoldings of 1-safe Petri nets to occurrence nets and event structures were introduced in [15], and have since been applied in a variety of areas from model checking to self-timed circuits and the fault diagnosis of communication networks. The unfoldings were given a universal characterisation a little later in [24] (or see [21]) and this had the useful consequence of providing a direct proof that unfolding preserved products and so many parallel compositions. There is an obstacle to an analogous universal characterisation of the unfolding of nets in which places/conditions hold with multiplicities: the symmetry between the multiple occurrences in the original net is lost in unfoldings to standard occurrence nets or event structures, and this spoils universality through non-uniqueness. However through the introduction of symmetry uniqueness up to symmetry obtains, and a universal characterisation can be regained [9].

We can illustrate the role symmetry plays in the unfolding of nets and hda's through a recent result relating event structures with symmetry to certain presheaves.⁶ Let \mathbb{P} be the category of finite elementary event structures (so essentially finite partial orders) with rigid maps. Form the presheaf category $\widehat{\mathbb{P}}$ which by definition is the functor category $[\mathbb{P}^{op}, \mathbf{Set}]$. From [27] we obtain that event structures with rigid maps (called 'strong' in [27]) embed fully and faithfully in $\widehat{\mathbb{P}}$ and are equivalent to those presheaves which are *separated* w.r.t. the Grothendieck topology with basis collections of jointly surjective maps in \mathbb{P} , and satisfy a further *mono* condition. Presheaves over $\widehat{\mathbb{P}}$ are thus a kind of generalised event structure.

There is clearly an inclusion functor $I : \mathbb{P} \hookrightarrow \widehat{\mathbb{P}}$ of finite elementary event structures into event structures with symmetry and rigid maps. Thus there is a functor $F : \mathcal{SE}_r \rightarrow \widehat{\mathbb{P}}$ taking an event structure with symmetry E to the presheaf $\mathcal{SE}_r(I(-), E)/\sim$. Event structures with symmetry yield more than just separated presheaves, and quite which presheaves they give rise to is not yet understood. But by restricting to event structures with symmetry $(E, l, r : S \rightarrow E)$ for which the symmetry is *strong*, in the sense that the mono $\langle l, r \rangle : S \rightarrow E \times E$ reflects consistency, we will always obtain nonempty separated presheaves. Let \mathcal{SSE}_r be the category of event structures with strong symmetry and rigid maps. Let $\text{Sep}(\widehat{\mathbb{P}})$ be the full subcategory of non-empty separated presheaves. So restricted, we obtain

⁶ The result is inspired by joint work with the Sydney Concurrency Group: Richard Buckland, Jon Cohen, Rob van Glabbeek and Mike Johnstone.

a functor $F : \mathcal{SSE}_r \rightarrow \widehat{\mathbb{P}}$ taking an event structure with strong symmetry E to the nonempty separated presheaf $\mathcal{SSE}_r(J(-), E)/\sim$. The functor F can be shown to have a right biadjoint, a functor G , producing an event structure with strong symmetry from a nonempty separated presheaf. The right biadjoint G is full and faithful (once account is taken of the the equivalence \sim on maps). (The existence of G relies on the event structures being consistent-countable—see the footnote, Section 2.) It shows how separated presheaves embed via a reflection fully and faithfully in event structures with symmetry:

$$\mathcal{SSE}_r \begin{array}{c} \xrightarrow{F} \\ \perp \\ \xleftarrow{G} \end{array} \text{Sep}(\mathbb{P}). \quad (\dagger)$$

The proof of the biadjunction has only been carried out for rigid maps, the reason why we have insisted that the maps of event structures in this section be rigid. (One could hope for a similar biadjunction without restricting F to strong symmetries.)

Higher-dimensional automata [7] are most concisely described as cubical sets, *i.e.* as presheaves over \mathbb{C} , a category of cube shapes of all dimensions with maps including *e.g.* ‘face’ maps, specifying how one cube may be viewed as a (higher-dimensional) face of another. We can identify the category of hda’s with the presheaf category $\widehat{\mathbb{C}}$. There are some variations in the choice of maps in \mathbb{C} , according to whether the cubes are oriented and whether degeneracy maps are allowed. For simplicity we assume here that the cubes are not oriented and have no degeneracy maps, so the maps are purely face maps. Roughly, then the maps of \mathbb{P} and \mathbb{C} only differ in that maps in \mathbb{P} fix the initial empty configuration whereas face maps in \mathbb{C} are not so constrained. By modifying the maps of \mathbb{P} to allow the initial configuration to shift under maps, we obtain a category \mathbb{A} into which both \mathbb{P} and \mathbb{C} include:

$$\mathbb{P} \hookrightarrow \mathbb{A} \xleftarrow{K} \mathbb{C}$$

Now we can construct a functor from $H : \mathbb{P} \rightarrow \widehat{\mathbb{C}}$; it takes p in \mathbb{P} to the presheaf $\mathbb{A}(K(-), J(p))$. Taking its left Kan extension over the Yoneda embedding of \mathbb{P} in $\widehat{\mathbb{P}}$ we obtain a functor

$$H_! : \widehat{\mathbb{P}} \rightarrow \widehat{\mathbb{C}}.$$

For general reasons [4], the functor $H_!$ has a right adjoint H^* taking an hda Y in $\widehat{\mathbb{C}}$ to the presheaf $\widehat{\mathbb{C}}(H(-), Y)$ in $\widehat{\mathbb{P}}$:

$$\widehat{\mathbb{P}} \begin{array}{c} \xrightarrow{H_!} \\ \perp \\ \xleftarrow{H^*} \end{array} \widehat{\mathbb{C}}. \quad (\ddagger)$$

We cannot quite compose the biadjunctions (\dagger) and the adjunction (\ddagger) because (\dagger) is only for separated presheaves. However restricting to hda’s which are separated, now w.r.t. a basis of jointly surjective maps in \mathbb{C} ,⁷ will ensure that they are

⁷ For a separated hda, cubes which share the same 1-dimensional edges must be equal (so ‘no ravioli’).

sent to separated presheaves over \mathbb{P} and so to event structures with symmetry. General Petri nets give rise to separated hda's (for example, with the 'self-concurrent individual token interpretation' of [7]). So we obtain a rather abstract construction of an unfolding of general nets to event structures with symmetry. Again, much more needs to be done, both mathematically in seeking a generalisation of the biadjunction (\dagger) to all event structures with symmetry, and in understanding unfoldings concretely so that they can be made amenable algorithmically.

Acknowledgement

I'm pleased to have this opportunity to acknowledge my appreciation and thanks to Gordon Plotkin for inspiration, guidance and friendship over the years.

I'm grateful to Marcelo Fiore, Martin Hyland, Lucy Saunders-Evans, Pawel Sobocinski, Sam Staton, Dominic Verity and the Sydney Concurrency Group for discussions and encouragement; Martin's advice has been especially valuable. I acknowledge the partial support of EPSRC grant GR/T22049/01. The paper was started on a visit to Aarhus University, and completed during an enjoyable visit to Macquarie University and NICTA, Sydney, Australia.

References

- [1] Abramsky, S., Jagadeesan, R., and Malacaria, P., Full Abstraction for PCF. *Information and Computation* vol. 163, 409–470, 2000.
- [2] Berry, G., Modèles complètement adéquats et stables des λ -calculs typés. Thèse de Doctorat d'Etat, Université de Paris VII, 1979.
- [3] Burroni, A., T-catégories. *Cahiers de topologie et géométrie différentielle*, XII 3, 1971.
- [4] Cattani, G.L., and Winskel, G., Profunctors, open maps and bisimulation. *MSCS*, 2005.
- [5] Cheng, E., Hyland, J.M.E., and Power, A.J., Pseudo-distributive laws. *ENTCS* 83, 2004.
- [6] Crazzolara, F., and Winskel, G., Composing Strand Spaces. *FSTTCS'02*, 2002.
- [7] Glabbeek, R.J.van, On the expressiveness of higher dimensional automata. *EXPRESS* 2004, *ENTCS* 128(2), 2005.
- [8] Doghmi, S.F., Guttman, J.D., and Thayer, F.J., Searching for shapes in cryptographic protocols. *TACAS'07*, 2007.
- [9] Hayman, J., and Winskel, G., The unfolding of general Petri nets. Forthcoming.
- [10] Johnstone, P., Sketches of an elephant, a topos theory compendium, vol.1. *OUP*, 2002.
- [11] Joyal, A., Nielsen, M., and Winskel, G., Bisimulation from open maps. *LICS '93 special issue of Information and Computation*, 127(2):164–185, 1996. Available as BRICS report, RS-94-7.
- [12] Kahn, G., and Plotkin, G.D., Concrete domains. *TCS*, 121(1& 2):187–277, 1993.
- [13] Koslowski, J., A monadic approach to polycategories. *Theory and Applications of Categories*, 14(7):125–156, 2005.
- [14] Mac Lane, S. *Categories for the Working Mathematician*. Springer, 1971.
- [15] Nielsen, M., Plotkin, G.D., and Winskel, G., Petri nets, event structures and domains. *TCS*, 13(1):85–108, 1981.

- [16] Nygaard, M., Domain theory for concurrency. PhD Thesis, University of Aarhus, 2003.
- [17] Nygaard, M., and Winskel, G., Domain theory for concurrency. *TCS* 316: 153–190, 2004.
- [18] Power, A.J., 2-Categories. *BRICS Lecture Notes, Aarhus University*, March, 1998.
- [19] Saunders-Evans, L., and Winskel, G., Event structure spans for non-deterministic dataflow. Proc. Express'06, ENTCS, 2006.
- [20] Saunders-Evans, L., Events with persistence. Forthcoming PhD, University of Cambridge Computer Laboratory.
- [21] Winskel, G., and Nielsen, M., Models for Concurrency. Handbook of Logic and the Foundations of Computer Science, vol. 4, pages 1-148, OUP, 1995.
- [22] Winskel, G., *Events in Computation*. PhD thesis, Univ. of Edinburgh, available from <http://www.cl.cam.ac.uk/users/gw104>, 1980.
- [23] Winskel, G., Event structure semantics of CCS and related languages. ICALP 82, Springer-Verlag LNCS 140, 1982. Extended version available from <http://www.cl.cam.ac.uk/users/gw104>.
- [24] Winskel, G., A new definition of morphism on Petri Nets. STACS'84: 140-150, 1984.
- [25] Winskel, G., Event structures. Invited lectures for the Advanced Course on Petri nets, September 1986. Springer LNCS, vol.255, 1987.
- [26] Winskel, G., An introduction to event structures. REX summerschool in temporal logic, Springer LNCS, vol.354, 1988.
- [27] Winskel, G., Event structures as presheaves—two representation theorems. CONCUR 1999. Springer LNCS, vol.1664, 1999.
- [28] Winskel, G., Relations in concurrency. Invited talk, LICS'05, 2005.

A Equivalence relations [10]

Assume a category with pullbacks. Let E be an object of the category. A *relation* on E is a pair of maps $l, r : S \rightarrow E$ for which l, r are *jointly monic*, *i.e.* for all maps $x, y : D \rightarrow S$, if $lx = ly$ and $rx = ry$, then $x = y$. Equivalently, if the category has binary products, a relation on E is a pair of maps $l, r : S \rightarrow E$ for which the mediating map $\langle l, r \rangle : S \rightarrow E \times E$ is monic. The relation is an *equivalence relation* in the category iff it is:

Reflexive: there is a (necessarily unique) map ρ such that

$$\begin{array}{ccc}
 & E & \\
 \text{id}_E \swarrow & \vdots \rho & \searrow \text{id}_E \\
 E & \xleftarrow{l} S \xrightarrow{r} & E
 \end{array}$$

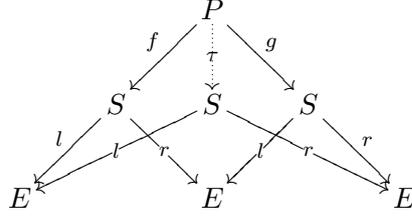
commutes;

Symmetric: there is a (necessarily unique) map σ such that

$$\begin{array}{ccc}
 & S & \\
 r \swarrow & \vdots \sigma & \searrow l \\
 E & \xleftarrow{l} S \xrightarrow{r} & E
 \end{array}$$

commutes;

Transitive: there is a (necessarily unique) map τ such that



commutes, where P, f, g is a pullback of r, l .

B Prime algebraic domains

Recall the definition of prime algebraic domain from [15,22,23]. We say a subset X of a partial order (D, \sqsubseteq) is *compatible*, written $X \uparrow$, iff it has an upper bound in D . A partial order (D, \sqsubseteq) is *consistent complete* iff whenever a subset $X \subseteq D$ is finitely compatible (*i.e.* any finite subset has an upper bound) it has a least upper bound $\bigsqcup X$. Note that any consistent complete partial order must have a least element \perp , the least upper bound of the empty subset. An element p of a consistent complete partial order is a *complete prime* iff for all compatible subsets X if $p \subseteq \bigsqcup X$ then $p \subseteq x$ for some $x \in X$. A *prime algebraic domain* is a bounded complete partial order such that for all $d \in D$

$$d = \bigsqcup \{p \sqsubseteq d \mid p \text{ is a complete prime}\} .$$

Say a prime algebraic domain is *finitary* iff every complete prime dominates only finitely many elements.

Theorem B.1 [15]

(i) For any event structure (E, Con, \leq) , the partial order $(\mathcal{C}(E), \subseteq)$ is a finitary prime algebraic domain.

(ii) For any finitary prime algebraic domain (D, \sqsubseteq) define (P, Con, \leq) where: P is the set of complete primes of D ; $X \in \text{Con}$ iff X is a finite subset of P bounded in D ; and $p \leq p'$ iff $p, p' \in P$ and $p \sqsubseteq p'$ in D . Then, (P, Con, \leq) is an event structure such that

$$\theta : (D, \sqsubseteq) \cong (\mathcal{C}(P), \subseteq)$$

is an isomorphism of partial orders where $\theta(d) = \{p \sqsubseteq d \mid p \text{ is a complete prime}\}$; its inverse takes a configuration $x \in \mathcal{C}(P)$ to $\bigsqcup x$.

C Stable families

So event structures can be obtained from finitary prime algebraic domains. One convenient way to construct finitary prime algebraic domains is from stable families [23]. The use of stable families facilitates constructions such as products and

pullbacks of event structures.

Definition C.1 A *stable family* (of finite configurations) comprises \mathcal{F} , a family of finite subsets, called *configurations*, satisfying:

Completeness: $Z \subseteq \mathcal{F} \ \& \ Z \uparrow \Rightarrow \bigcup Z \in \mathcal{F}$;

Coincidence-freeness: For all $x \in \mathcal{F}$, $e, e' \in x$ with $e \neq e'$,

$$(\exists y \in \mathcal{F}. y \subseteq x \ \& \ (e \in y \iff e' \notin y)) ;$$

Stability: $\forall Z \subseteq \mathcal{F}. Z \neq \emptyset \ \& \ Z \uparrow \Rightarrow \bigcap Z \in \mathcal{F}$.

For $Z \subseteq \mathcal{F}$, we write $Z \uparrow$ to mean compatibility in \mathcal{F} w.r.t. the inclusion order. We call members of the set $\bigcup \mathcal{F}$, the *events* of \mathcal{F} .

A stable family of finite configurations provides a representation of the finite elements of a finitary prime algebraic domain.⁸ Configurations of stable families each have their own local order of causal dependency, so their own prime sub-configurations generated by their events. We can build an event structure by taking the events of the event structure to comprise the set of all prime configurations of the stable family. (The prime configurations corresponding to complete primes of the domain.)

Proposition C.2 Let x be a configuration of a stable family \mathcal{F} . For $e, e' \in x$ define

$$e' \leq_x e \text{ iff } \forall y \in \mathcal{F}. y \subseteq x \ \& \ e \in y \Rightarrow e' \in y.$$

When $e \in x$ define the *prime configuration*

$$[e]_x = \bigcap \{y \in \mathcal{F} \mid y \subseteq x \ \& \ e \in y\} .$$

Then \leq_x is a partial order and $[e]_x$ is a configuration such that

$$[e]_x = \{e' \in x \mid e' \leq_x e\}.$$

Moreover the configurations $y \subseteq x$ are exactly the down-closed subsets of \leq_x .

Proposition C.3 Let \mathcal{F} be a stable family. Then, $\text{Pr}(\mathcal{F}) =_{\text{def}} (P, \text{Con}, \leq)$ is an

⁸ There are some minor differences with stable families as originally introduced in Definition 1.1 of [23]. Here it is convenient to restrict to finite configurations, obviating the ‘finitary’ axiom of [23], weaken to ‘completeness’ rather than ‘coherence,’ and assume the family is ‘full,’ that every event appears in some configuration. The expanded article [23] remains a good reference for the proofs of the results here in the appendix.

event structure where:

$$\begin{aligned} P &= \{[e]_x \mid e \in x \ \& \ x \in \mathcal{F}\} , \\ Z \in \text{Con} &\text{ iff } Z \subseteq P \ \& \ \bigcup Z \in \mathcal{F} \text{ and,} \\ p \leq p' &\text{ iff } p, p' \in P \ \& \ p \subseteq p' . \end{aligned}$$

This proposition furnishes a way to construct an event structure with events the prime configurations of a stable family. In fact we can equip the class of stable families with maps. The definitions are just copies of those for event structures. For example, a (total) map of stable families $f : \mathcal{F} \rightarrow \mathcal{G}$ is a function $f : \bigcup \mathcal{F} \rightarrow \bigcup \mathcal{G}$ such that for all configurations $x \in \mathcal{F}$ its direct image $fx \in \mathcal{G}$ for which

$$\text{if } e_1, e_2 \in x \text{ and } f(e_1) = f(e_2), \text{ then } e_1 = e_2.$$

The open maps for stable families, specified w.r.t. (the families of configurations of) finite elementary event structures as paths, are characterised just as in Proposition 2.5 as rigid maps satisfying a further lifting property. We shall concentrate on the category of stable families with total maps, \mathcal{Fam} .

We shall make use of an important adjunction between event structures and stable families. The configurations of an event structure form a stable family. The corresponding functor “inclusion” functor $\mathcal{C}^o : \mathcal{E} \rightarrow \mathcal{Fam}$ takes an event structure E to the stable family $\mathcal{C}^o(E)$, and a map $f : E \rightarrow E'$ in \mathcal{E} to the map $f : \mathcal{C}^o(E) \rightarrow \mathcal{C}^o(E')$ between the stable families of their finite configurations. The functor $\text{Pr} : \mathcal{Fam} \rightarrow \mathcal{E}$ acts on objects as described in the proposition above, producing an event structure out of the prime configurations of a stable family. It takes a map $f : \mathcal{F} \rightarrow \mathcal{G}$ in \mathcal{Fam} to the map $\text{Pr} f : \text{Pr}(\mathcal{F}) \rightarrow \text{Pr}(\mathcal{G})$ given by $\text{Pr} f([a]_x) = [f(a)]_{fx}$ whenever $a \in x$ and $x \in \mathcal{F}$.

Theorem C.4 *Let $\mathcal{C}^o : \mathcal{E} \rightarrow \mathcal{Fam}$ and $\text{Pr} : \mathcal{Fam} \rightarrow \mathcal{E}$ be the functors defined above. Then $\mathcal{C}^o \dashv \text{Pr}$. The unit η of the adjunction has components $\eta_A : A \rightarrow \text{Pr} \mathcal{C}^o(A)$ given by $\eta_A(a) = [a]$ where a is an event of A —recall $[a] = \{a' \in A \mid a' \leq a\}$. The counit is the natural transformation with components $\epsilon_{\mathcal{F}} : \mathcal{C}^o \text{Pr}(\mathcal{F}) \rightarrow \mathcal{F}$ given by $\epsilon_{\mathcal{F}}([b]_x) = b$ for $x \in \mathcal{F}$ and $b \in x$.*

The unit is a natural isomorphism. The component of the counit $\epsilon_{\mathcal{F}}$ at stable family \mathcal{F} satisfies:

$$\forall x \in \mathcal{F}, y \in \mathcal{C}^o \text{Pr}(\mathcal{F}). \ \epsilon_{\mathcal{F}} y = x \text{ iff } y = \{[b]_x \mid b \in x\} .$$

An almost identical story holds with respect to other maps, rigid and partial; the “inclusion” functor from the corresponding category of event structures to the corresponding category of stable families has a right adjoint, defined as Pr above, but with the obvious slight adjustments to the way it acts on maps.

Lemma C.5 *Components $\epsilon_{\mathcal{F}}$ of the counit of the adjunction $\mathcal{C}^o \dashv \text{Pr}$ are open. The functor Pr preserves open maps.*

Proof. There are direct proofs from the definitions and Theorem C.4. Alternatively, there are general diagrammatic proofs. See Lemma 6 of [11] for the openness of the counit, and Lemma 2.5 of [4] to show Pr preserves opens. \square

C.1 Products and pullbacks

The adjunction between event structures and stable families is a coreflection in the sense that the unit η is a natural isomorphism. The fact that the adjunction is a coreflection, has the useful consequence of allowing the calculation of limits in event structures from the more easily constructed limits in stable families. For example, products of event structures (the specific event structures of this article) are hard to define directly. It is however straightforward to define products of stable families [23], and from them to obtain the products of event structures by using Pr .

Let \mathcal{F}_1 and \mathcal{F}_2 be stable families with events E_1 and E_2 , respectively. Their product in \mathcal{Fam} , the stable family $\mathcal{F}_1 \times \mathcal{F}_2$, will have events comprising pairs in $E_1 \times E_2$, the product of sets with projections π_1 and π_2 :

$$\begin{aligned}
x \in \mathcal{F}_1 \times \mathcal{F}_2 \text{ iff} \\
& x \text{ is a finite subset of } E_1 \times E_2, \\
& \pi_1 x \in \mathcal{F}_1 \ \& \ \pi_2 x \in \mathcal{F}_2, \\
& \forall e, e' \in x. \pi_1(e) = \pi_1(e') \text{ or } \pi_2(e) = \pi_2(e') \Rightarrow e = e', \text{ and} \\
& \forall e, e' \in x. e \neq e' \Rightarrow \exists y \subseteq x. \pi_1 y \in \mathcal{F}_A \ \& \ \pi_2 y \in \mathcal{F}_B \ \& \ (e \in y \iff e' \notin y).
\end{aligned}$$

Notice that the third condition says that x is a bijection between $\pi_1 x$ and $\pi_2 x$. The projections of the product of stable families, $\mathcal{F}_1 \times \mathcal{F}_2$, are given by the projections π_1 and π_2 , from the set of pairs of events, (e_1, e_2) , which appear in a configuration of the product to the left, e_1 , and right component, e_2 , respectively.

Right adjoints preserve limits, and so products in particular. Consequently we obtain a product of event structures E_1 and E_2 by first regarding them as stable families $\mathcal{C}^o(E_1)$ and $\mathcal{C}^o(E_2)$, and then producing the event structure from the product $\mathcal{C}^o(E_1) \times \mathcal{C}^o(E_2), \pi_1, \pi_2$ of the stable families. Indeed we will define the product of event structures, $E_1 \times E_2 = \text{Pr}(\mathcal{C}^o(E_1) \times \mathcal{C}^o(E_2))$. Its projections are obtained as $\eta_{E_1}^{-1} \circ (\text{Pr } \pi_1)$ and $\eta_{E_2}^{-1} \circ (\text{Pr } \pi_2)$, which take $[(e_1, e_2)]_x$ to e_1 and e_2 , respectively.

Products in the category \mathcal{Fam}_r , stable families with rigid maps, can be obtained from products in \mathcal{Fam} by restricting to those configurations on which the projections are rigid. More fully, suppose \mathcal{F}_1 and \mathcal{F}_2 are stable families. Their ‘rigid’ product is obtained as the stable family consisting of those configurations $x \in \mathcal{F}_1 \times \mathcal{F}_2$, the product in \mathcal{Fam} , on which the projections are rigid, *i.e.* for which

$$\begin{aligned}
& \forall z \in \mathcal{F}_1. z \subseteq \pi_1 x \Rightarrow \exists x' \in \mathcal{F}_1 \times \mathcal{F}_2. x' \subseteq x \ \& \ \pi_1 x' = z \text{ and} \\
& \forall z \in \mathcal{F}_2. z \subseteq \pi_2 x \Rightarrow \exists x' \in \mathcal{F}_1 \times \mathcal{F}_2. x' \subseteq x \ \& \ \pi_2 x' = z.
\end{aligned}$$

For maps $f_1 : \mathcal{G} \rightarrow \mathcal{F}_1$ and $f_2 : \mathcal{G} \rightarrow \mathcal{F}_2$ in \mathcal{Fam} , there is unique mediating map to the product, the pair $\langle f, g \rangle : \mathcal{G} \rightarrow \mathcal{F}_1 \times \mathcal{F}_2$. If both f and g are rigid, then so is

$\langle f, g \rangle$. The analogous property holds for event structures.

Once we have products of event structures in \mathcal{E} , pullbacks in \mathcal{E} are obtained by restricting products to the appropriate equalizing set. Pullbacks in \mathcal{E} can also be constructed via pullbacks in \mathcal{Fam} , in a similar manner to the way we have constructed products in \mathcal{E} . We obtain pullbacks in \mathcal{Fam} as restrictions of products. Suppose $f_1 : \mathcal{F}_1 \rightarrow \mathcal{G}$ and $f_2 : \mathcal{F}_2 \rightarrow \mathcal{G}$ are maps in \mathcal{Fam} . Let E_1, E_2 and C be the sets of events of $\mathcal{F}_1, \mathcal{F}_2$ and \mathcal{G} , respectively. The set $P =_{\text{def}} \{(e_1, e_2) \mid f_1(e_1) = f_2(e_2)\}$ with projections π_1, π_2 to the left and right, forms the pullback, in the category of sets, of the functions $f_1 : E_1 \rightarrow C, f_2 : E_2 \rightarrow C$. We obtain the pullback in \mathcal{Fam} of f_1, f_2 as the stable family \mathcal{P} , consisting of those subsets of P which are also configurations of the product $\mathcal{F}_1 \times \mathcal{F}_2$ —its associated maps are the projections π_1, π_2 from the events of \mathcal{P} .

Pullbacks in \mathcal{Fam}_r are constructed analogously. The coreflection between \mathcal{E}_r and \mathcal{Fam}_r enables us to construct products and pullbacks in \mathcal{E}_r .