Event Structures with Symmetry
(Extended Abstract)

Glynn Winskel [1]

University of Cambridge Computer Laboratory, England

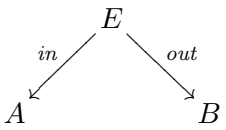
Abstract

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

Keywords: Event structures, symmetry, pseudo monads, spans, semantics of processes, Petri net unfolding.

1 Introduction

In the paper introducing event structures [7] 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 [9] for a summary). This slogan stands for an attempt to push the methodology of domain theory and denotational semantics into the areas of interactive/distributed/concurrent computation, where presently more syntactic, operational or more informal methodologies rule. Certain generalized relations (profunctors [2]) 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 [8,16,11]. A span of event structures is typically of the form



[1] Email: gw104@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 a monads on a fundamental category of event structures. 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.

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 is present informally in the model of strand spaces. 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 so reducing the problem to the unfolding in [7]. 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 [2]. The computation paths of an event structure, its configurations, are ordered by inclusion. In contrast the paths of presheaf models can be 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 monads, from

which we can then obtain broader kinds of span.

2 Event structures

Event structures [7,12,14,15] are a model of computational processes. They represent a process 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 $(E, \mathrm{Con}, \leq)$, consisting of a countable 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

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

The events are to be thought of as event occurrences; in any history an event is to appear at most once. A configuration is a set of events which have occurred by some stage in a process. According to our understanding of the consistency predicate 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 *(finite) configurations*, $\mathcal{C}(E)$, of an event structure E consist of those finite subsets $x \subseteq E$ which are

Consistent: $x \in \mathrm{Con}$ and

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

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. For an event e the set $\{e' \in E \mid e' \leq e\}$ 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, because an event determines its whole causal history. One closely related and more versatile model is that of stable families, described in the Appendix.

Let E and E' be event structures. A *partial map* of event structures $f : E \rightharpoonup E'$

is a partial function on events $f : E \rightharpoonup E'$ such that for all configurations x of E its direct image fx is a configuration of E' for which

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

The 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. Maps of event structures compose as partial functions.

We will say the map is *total* iff the function f is total, and *rigid* iff it is total and for all configurations x of E and y of E'

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

(The configuration z is necessarily unique.)

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; 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 [6].)

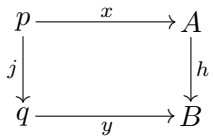
Here we concentrate on the category of event structures with total maps.

Definition 2.1 Write $\mathcal{E}$ for the category of event structures with total maps. (In future by a map of event structures we will mean a total map.)

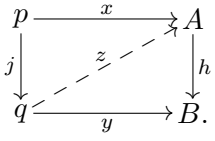
Proposition 2.2 *The category $\mathcal{E}$ of event structures with total maps of event structures has products and pullbacks (though no terminal object).*

In defining symmetries on event structures we will make use of open maps w.r.t. finite elementary event structures (*i.e.*, finite event structures in which all subsets are consistent) as the particular choice of paths [5].

Say a map $h : A \to B$, between event structures A and B, is *open* iff for all maps $j : p \to q$ between finite elementary event structures, any commuting square



can be split into two commuting triangles



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 are a generalisation of functional bisimulations, known from transition systems.

Proposition 2.3 *Open maps are rigid.*

3 Event structures with symmetry

A symmetry on an event structure expresses when and how two finite configurations are similar. Such similarity should form an equivalence relation. If two configurations are similar then so should their pasts (restrictions to subconfigurations) and futures (extensions to larger configurations) be similar. These properties are captured, somewhat abstractly, by the following definition.

Definition 3.1 An *event structures with symmetry* (E, l, r) comprises an event structure E together with *open* maps $l : S \to E$ and $r : S \to E$ from a common event structure S such that the map $\langle l, r\rangle : S \to E \times E$ is an equivalence relation (*i.e.*the map $\langle l, r\rangle$ is monic and satisfies the standard diagramatic properties of reflexivity, symmetry and transitivity [4]).

In the above definition, the role of the symmetry l, r is to specify a relation of similarity between the finite configurations of the event structure E. One can characterise such symmetries more concretely as corresponding to certain families of isomorphisms between finite configurations of E.

Definition 3.2 A *isomorphism family* of an event structure E consists of a family $\mathcal{S}$ of bijections

$$\theta : x \cong y$$

between pairs of finite configurations of E such that:

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

(ii) for $\theta : x \cong y$ in $\mathcal{S}$ 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 $\mathcal{S}$.

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

Note that (i) implies that the symmetric forms of (ii) and (iii) also hold. Note too that (ii) implies that the bijections in the family $\mathcal{S}$ 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.

Proposition 3.3 *Let E be an event structure.*

(i) A symmetry on $\langle l, r\rangle : S \to E \times E$ determines an isomorphism family $\mathcal{S}$: the family $\mathcal{S}$ comprises the image under $\langle l, r\rangle$ of all the finite configurations of S.

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

The operations of (i) and (ii) are mutually inverse.

Through the addition of symmetry event structures can represent a much richer class of 'path categories' [2] 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 composing the inclusions and allowed isomorphisms; by property (ii) in Definition 3.2 any such map factors uniquely as an allowed isomorphism followed by an inclusion.

Example 3.4 Any event structure E can be identified with the event structure with the identity symmetry $(E, \mathrm{id}_E, \mathrm{id}_E)$. Its isomorphism family consists of all identities $\mathrm{id}_x : x \cong x$ on finite configurations $x \in \mathcal{C}(E)$.

Example 3.5 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 trivlal 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 $l : S \to E$ and $r : S \to E$. Then $\varpi =_{\mathrm{def}} (\omega, l, r)$ 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 set.

Example 3.6 Let $E = (E, l : S \to E, r : S \to E)$ be an event structure with symmetry. Define an event structure with symmetry $!E = (E_!, l_! : S_! \to E_!, r_! : S_! \to E_!)$ 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

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

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 $\mathrm{Con}_{S_!}$ iff

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

The finite configurations of $E_!$ correspond to tuples (or indexed families) $\langle x_i \rangle_{i \in I}$ of 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 $i \in I$.

There is an isomorphism of event structures with symmetry $\varpi \cong !1$, where 1 is the event structure with a single event.

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 $f : (A, l_A, r_A) \to (B, l_B, r_B)$ is a map of event structures $f : A \to B$ such that there is a (necessarily unique) map of event structures $h : S_A \to S_B$ ensuring

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

Maps between event structures with symmetry compose as maps of event structures and share the same identity maps. We define $\mathcal{ES}$ to be category of event structures with symmetry.

We explore properties of the category $\mathcal{ES}$. It is more fully described as category enriched over 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 $\mathcal{ES}$ which are often very simple examples of the (pseudo- and bi-) constructions of 2-categories.

Definition 4.1 Let $f, g : (A, l_A, r_A) \to (B, l_B, r_B)$ 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 $h : A \to S_B$ such that

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

Proposition 4.2 *The relation $\sim$ is an equivalence relation on maps $\mathcal{ES}(A, B)$ between event structures with symmetry A and B. The relation $\sim$ respects composition in the sense that if $f \sim g$ then $h \circ f \sim h \circ g$ and $f \circ k \sim g \circ k$, for composable maps h and k.*

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

Definition 4.3 Let A and B be event structures with symmetry. An *equivalence* from A to B is a pair of maps $f : A \to B$ and $g : B \to A$ such that $f \circ g \sim \mathrm{id}_B$ and $g \circ f \sim \mathrm{id}_A$; then we say A and B are equivalent.

Proposition 4.4 *The category $\mathcal{ES}$ has products.*

The category $\mathcal{ES}$ does not have a terminal object. However, the event structure with symmetry ϖ defined in Example 3.4 satisfies an appropriately weakened property:

Proposition 4.5 *For any event structure with symmetry A there is a map $f : A \to \varpi$ in $\mathcal{ES}$ and moreover for any two maps $f, g : A \to \varpi$ we have $f \sim g$.*

The category $\mathcal{ES}$ does not have pullbacks and equalizers. However:

Proposition 4.6
(i) Let $f, g : A \to B$ be two maps between event structures with symmetry. There is an event structure with symmetry E and map $e : E \to A$ such that $f \circ e \sim g \circ e$ which satisfies the further property that for any event structure with symmetry E' and map $e' : E' \to A$ such that $f \circ e' \sim g \circ e'$, there is a unique map $h : E' \to E$ such that $e' = e \circ h$.
(ii) Let $f : A \to C$ and $g : B \to C$ be two maps between event structures with symmetry. There is an event structure with symmetry D and maps $p : D \to A$ and $q : D \to B$ such that $f \circ p \sim g \circ q$ which satisfies the further property that for any event structure with symmetry D' and maps $p' : D' \to A$ and $q' : D' \to B$ such that $f \circ p' \sim g \circ q'$, there is a unique map $h : D' \to D$ such that $p' = p \circ h$ and $q' = q \circ h$.

The defining property (i) above is a special, very simple, case of *inserters* in pie limits [10]. 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.

Because $\mathcal{ES}$ is enriched over equivalence relations, the appropriate functors on $\mathcal{ES}$ preserve the equivalence $\sim$ on the homsets. Ordinary natural transformations between such functors will automatically preserve $\sim$ because composition does.

5 Functors and pseudo monads

Certain functors on $\mathcal{E}$, the category of event structures, straightforwardly induce functors on $\mathcal{ES}$, the enriched category of event structures with symmetry. Suppose a functor $F : \mathcal{E} \to \mathcal{E}$ on event structures preserves pullbacks and open maps. Then it will induce a functor on $\mathcal{ES}$ which takes an event structure with symmetry $(A, l : S \to A, r : S \to A)$ to an event structure with symmetry $(F(A), F(l), F(r))$ and a map $f : (A, l_A, r_A) \to (B, l_B, r_B)$ to $F(f)$; it is easy to check that F preserves $\sim$ on homsets.

Similarly a functor on several arguments $F : \mathcal{E} \times \cdots \times \mathcal{E} \to \mathcal{E}$ which preserves pullbacks and open maps will induce a functor on event structures with symmetry respecting $\sim$ on homsets.

Simple parallel composition
For example, consider the functor $\|\colon \mathcal{E} \times \mathcal{E} \to \mathcal{E}$ which given two event structures puts them in parallel. Let $(A, \mathrm{Con}_A, \leq_A)$ and $(B, \mathrm{Con}_B, \leq_B)$ 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 $\{a \mid (0, a) \in C\} \in \mathrm{Con}_A$ and $\{b \mid (1, b) \in C\} \in \mathrm{Con}_B$. The operation extends to a functor—put the two maps in parallel. It is not hard to check that the functor $\|$ preserves pullbacks and open maps. Consequently it induces a functor $\|\colon \mathcal{ES} \times \mathcal{ES} \to \mathcal{ES}$ which preserves $\sim$ on homsets.

Sum
Similarly, the coproduct or sum of two event structures extends to the sum of event structures with symmetry. Let $(A, \mathrm{Con}_A, \leq_A)$ and $(B, \mathrm{Con}_B, \leq_B)$ 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 $C_0 \in \mathrm{Con}_A$ such that $C = \{(0, a) \mid a \in C_0\}$ or there is $C_1 \in \mathrm{Con}_B$ such that $C = \{(1, a) \mid a \in C_1\}$.

That $\mathcal{ES}$ is enriched over equivalence relations ensures that it supports the definitions pseudo functors and pseudo natural transformations, which here parallel those of functor and natural transformation, but with equality replaced by $\sim$. In the same spirit a pseudo monad on $\mathcal{ES}$ satisfies variants of the usual monad laws but expressed in terms of $\sim$ rather than equality (we can ignore the extra coherence conditions [3] 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.

The examples are based on constructions we have seen earlier.

5.1 The copying pseudo monad

The copying operation ! of Example 3.6 extends to a functor on $\mathcal{ES}$. Let $f : A \to B$ be a map of event structures with symmetry. Define $!f :!A \to !B$ 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 $\eta^!_E : E \to !E$ 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 $[\_, \_] : \omega \times \omega \to \omega$ which we assume is injective. The component of the multiplication $\mu^!_E :!!E \to !E$ 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.)

5.2 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 $\eta^*_E : E \to E$ acts so $\eta^*_E(e) = (0, e)$ for all events $e \in E$—so taking E to its copy.

The component of the multiplication $\mu^*_E : (E_*)_* \to E_*$ 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 [13]. It readily generalizes to accommodate symmetry and reappears as the Kleisli category of $(\_)_*$.

Definition 5.1 Let (A, l_A, r_A) and (B, l_B, r_B) be event structure with symmetry. A *partial map* of event structures with symmetry $f : (A, l_A, r_A) \rightharpoonup (B, l_B, r_B)$ consists of a partial map of event structures $f : A \rightharpoonup B$ for which there is a (necessarily unique) partial map of event structures $h : S_A \rightharpoonup S_B$ ensuring

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

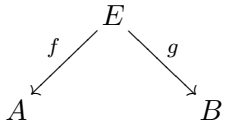
Partial maps of event structures with symmetry form a category; they compose as partial maps of event structures and share the same identity maps. We can define an equivalence relation $\sim$ on partial maps of event structures with symmetry by the obvious analogue of Definition 4.1. The category is enriched over equivalence relations. (The full subcategory of event structures with identity symmetry is isomorphic to the category of event structures with partial maps.)

Proposition 5.2 *The Kleisli category of the pseudo monad $(-)_*$ and the category of event structures with symmetry and partial maps are to biequivalent (regarding categories enriched over equivalence relations as 2-categories).*

6 Spans

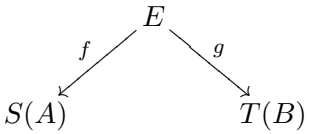
Because $\mathcal{ES}$ has bipullbacks in the sense of Proposition 4.6 we can imitate the standard construction of the bicategory of spans to produce a bicategory $\text{Span}_{\mathcal{ES}}$. Its objects are event structures with symmetry. Its maps $\text{Span}_{\mathcal{ES}}(A, B)$, from A to

B, are spans



composed using the bipullbacks of of Proposition 4.6(ii). Its 2-cells, maps in $\mathrm{Span}_{\mathcal{ES}}(A, B)$, are the maps between the vertices of two spans making the obvious triangles commute. $\mathrm{Span}_{\mathcal{ES}}$ has a tensor and function space given by the product of $\mathcal{ES}$.

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 $\mathcal{ES}$ such a process is rather restricted; from a computational view the process 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 $\mathcal{ES}$ 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, and form a bicategory, we require that S and T satisfy several requirements, which we indicate here:

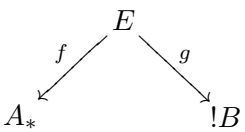
- in order to lift to pseudo comonads and monads on spans, S and T should be 'cartesian' pseudo monads, now w.r.t. bipullbacks (adapting [1]);
- in order to obtain a comonad-monad distributive law for the liftings of S and T to spans it suffices to have a distributive law for S and T, with commutativity up to $\sim$, with extra bipullback conditions on two of the four diagrams.

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

7 Applications

We briefly discuss some applications.

The particular bicategory of spans



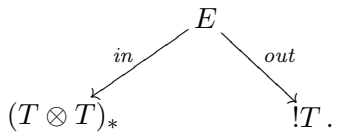
is already quite an interesting framework for the semantics of higher-order processes. It supports types including:

- Prefix types $\bullet!T$: 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 obvious generalization of the binary functor described in Section 5. Sum types may also be written $a_1T_1 + \cdots + a_nT_n$ when the indexing set is finite.
- Tensor types $T_1 \otimes T_2$: the product w.r.t. partial maps of event structures with symmetry.
- Function types $T_1 \multimap T_2$: formed as the product $(T_1)_* \times !T_2$ in $\mathcal{ES}$.
- Recursively defined types.

The types describe the events and basic causalities of a process, and in this sense are the *event types* or *causal types* of a process. For example, the type of a form of higher-order CCS is expressible as the recursive type:

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

The events of a h.o.CCS process are either process events following a τ-event, 'concretion' events following an output synchronizations $\bar{a}$, or 'abstraction' events following an input synchronization a. As an example of a higher order process, consider parallel composition in h.o.CCS. It will denote a span



The role of ! on the output is to permit a process to perform the events specified by its type countably many times. The types support definitions by cases on the form of events, a style of definition which breaks away from traditional 'interleaving' approaches to concurrency.

This work begs the question of the extent to which other kinds of maps and spans of event structures with symmetry can be obtained by Kleisli constructions starting from the category of event structures with symmetry, perhaps with rigid maps rather than total maps of event structures as the starting point, along the lines of [16]—to what extent can the 'persistent' events of [16] be realized through symmetry? Another direct application is to the unfolding of Petri nets with multiple tokens. It is intended to treat these applications more completely in the full paper.

Appendix: Stable families

The use of stable families facilitates definitions on event structures.

Definition A *stable family* comprises $(E, \mathcal{F})$ where E is a set of *events* and $\mathcal{F}$ is a family of finite subsets of E, 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, *i.e.*

$$\exists x \in \mathcal{F} \forall z \in Z.\ z \subseteq x\ .$$

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 sub-configurations of the stable family.

Definition and Proposition 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 and Definition Let $(E, \mathcal{F})$ be a stable family. Then, $\Pr(E) =_{def} (P, \mathrm{Con}, \leq)$ is an event structure where:

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

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 the same as those for event structures).

The configurations of an event structure form a stable family, so in this sense event structures are included in stable families. With respect to any of the maps (rigid, total or partial), the "inclusion" functor from the category of event structures to the category of stable families has a right adjoint, which on objects is the construction we have just given, producing an event structure from a stable family. The products w.r.t. total and partial maps are hard to define directly on the event structures of this article. It is however straightforward to define the products of stable families [13]. Right adjoints preserve limits, and so products in particular. Consequently we obtain products of event structures by first regarding them as stable families, and then producing the event structure from the product of the stable families. Pullbacks of event structures are obtained by restricting products to the appropriate equalizing set.

Acknowledgement

I'm grateful for discussions with Marcelo Fiore, Martin Hyland and Pawel Sobocinski. Birthday greetings and thanks to Gordon Plotkin for inspiration, guidance and friendship over the years.

References

[1] Burroni, A., T-catégories. *C*ahiers de topologie et géométrie différentielle, XII 3, 1971.

[2] Cattani, G.L., and Winskel, G., Profunctors, open maps and bisimulation. MSCS, 2005.

[3] Cheng, E., Hyland, J.M.E., and Power, A.J., Pseudo-distributive laws. ENTCS 83, 2004.

[4] Johnstone, P., Sketches of an elephant, a topos theory compendium, vol.1. *OUP*, 2002.

[5] 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.

[6] Kahn, G., and Plotkin, G.D., Concrete domains. *TCS*, 121(1& 2):187–277, 1993.

[7] Nielsen, M., Plotkin, G.D., and Winskel, G., Petri nets, event structures and domains. *TCS*, 13(1):85–108, 1981.

[8] Nygaard, M., Domain theory for concurrency. PhD Thesis, University of Aarhus, 2003.

[9] Nygaard, M., and Winskel, G., Domain theory for concurrency. *TCS* 316: 153–190, 2004.

[10] Power, A.J., and Robinson, E.P., A characterization of pie limits. *Math. Proc. Camb. Phil.Soc,* 33-47, 1991.

[11] Saunders-Evans, L., and Winskel, G., Trace on event structures. Proc. Express'06, ENTCS, 2006.

[12] Winskel, G., *Events in Computation*. PhD thesis, Univ. of Edinburgh, available from http://www.cl.cam.ac.uk/users/gw104, 1980.

[13] 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.

[14] Winskel, G., Event structures. Invited lectures for the Advanced Course on Petri nets, September 1986. Springer LNCS, vol.255, 1987.

[15] Winskel, G., An introduction to event structures. REX summerschool in temporal logic, Springer LNCS, vol.354, 1988.

[16] Winskel, G., Relations in concurrency. Invited talk, LICS'05, 2005.