Geometric Logic, Causality and Event Structures

Jeremy Gunawardena

26 February 1991

Abstract

The conventional approach to causality is based on partial orders. Without additional structure, partial orders are only capable of expressing AND causality. In this paper we investigate a syntactic, or logical, approach to causality which allows other causal relationships, such as OR causality, to be expressed with equal facility. In earlier work, [3], we showed the benefits of this approach by giving a causal characterisation, in the finite case, of Milner's notion of confluence in CCS. This provides the justification for the more systematic study of causality, without finiteness restrictions, which appears here. We identify three general principles which a logic of causality should satisfy. These principles summarise some basic intuitions about events and causality. They lead us to geometric logic - the "logic of finite observations" - as a candidate for a logic of causality. We introduce the formalism of geometric automata based on this choice; a geometric automaton is a set E together with a pair of endomorphisms of the free frame (locale) generated by E. Our main result is to show that Winskel's general event structures are a special case of geometric automata. This is analogous to the transition from topological data (sets of points) to algebraic structures (lattices of open subsets) in "pointless topology", [6]. This result links our ideas on causality with Winskel's theory of events in computation; it provides a syntax for describing event structures and it opens the way to giving a causal interpretation of event structure phenomena. We show further that geometric automata give rise to domains of configurations which generalise the event domains of Winskel and Droste.

1 Introduction

In this paper we develop a syntactic approach to causality. We seek to find a language (a logic or algebra) in which causal relationships can be described and to embody this in a formalism for reasoning about reactive systems.

Causality is conventionally represented by a partially ordered set (poset). The poset whose Hasse diagram is shown in Figure 1 can be rendered into syntactic form as shown in Figure 2. The arrow symbol, $\rightarrow$, can be read as "is-caused-by". The special symbol T, for the moment un-interpreted, but with obvious logical connotations, indicates that the corresponding event is initial in the partial order. $\wedge$ has its usual meaning of logical AND. It is clear that any partial order could be translated into such a table, possibly infinite, using only the connective AND. We summarise this by saying that posets express only AND causality.

One advantage of the syntactic approach is that we can easily express other forms of causality, such as OR causality. We are accustomed to think of AND and OR as dual connectives of comparable importance. The table below seems as meaningful as the one in Figure 2.

a	$\rightarrow$	T
b	$\rightarrow$	T
c	$\rightarrow$	$a \vee b$

(We shall define an operational semantics for such tables in §3; the behaviour of this one should be reasonably clear.) In an earlier paper we used tables like these to give a causal characterisation, [3, Theorem 1.1], of Milner's notion of confluence in CCS, [7, Chapter 11]. This application provides a justification for the more systematic treatment of causality in the present paper. The reader is referred to [3, §1] for a discussion of OR causality and its significance in concurrency theory.

The tables which we constructed above in an informal way can be seen more formally as pairs (E, ρ) where E is some set of events and $\rho : E \rightarrow \mathcal{L}(E)$ is a function ("is caused by") from E to some logic, $\mathcal{L}(E)$ generated by the symbols in E. It seems to require at least the binary connectives AND, and OR and the unary connective T. From a logical viewpoint we shall be concerned with the model theory rather than the proof theory and we will work exclusively with the Lindenbaum algebra of $\mathcal{L}$. However, we find it convenient to continue to use the word "logic" in preference to "algebra" partly to avoid the associations with "process algebra". The syntax which is developed here allows us to describe causal relationships; it does not provide us with process constructors.

One objective of this paper is to understand the requirements for a logic of causality. In §2 we state three general principles which a logic of causality should satisfy. These principles summarise our foundational intuitions about events and causality. They limit the possible choices of logic and allow us to measure the expressibility of any particular logic.

In our earlier paper, [3], we restricted attention to the case of finitely many events and took the path of least resistance by choosing classical Boolean logic as the logic of causality. This gave rise to "tables" which we called causal automata. In the infinite case the choice of logic must be made with greater care and throws up some interesting problems which are still not completely resolved.

In §3 we show that geometric logic - the "logic of finite observations", [12] - is a suitable candidate for a logic of causality and we introduce a formalism based on this called geometric automata. Geometric logic, or its algebrisation, the theory of frames (locales), is the subject matter of "pointless topology", [6]. The significance of this to computer science was first

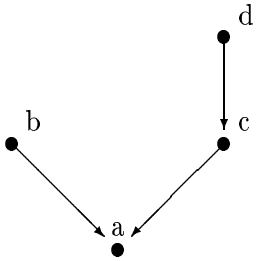

Figure 1: Hasse diagram of a poset

$$\begin{array}{lcl} a & \rightarrow & b \wedge c \\ b & \rightarrow & \mathsf{T} \\ c & \rightarrow & d \\ d & \rightarrow & \mathsf{T} \end{array}$$

Figure 2: Syntactic version of the poset

noted by Smyth, [9], and followed up in Abramsky's programme, [1]. We define the concepts needed for the present paper but refer to Vickers' introductory treatment, [12], and Johnstone's treatise, [5], for details. A geometric automaton is simply a set, E, together with a pair of endomorphisms of the free frame generated by E.

A key result in §3 is Proposition 3.1 which gives a topological interpretation of the free frame. This is the essential ingredient in proving our main result, in §4, that Winskel's general event structures, [14], are a special case of geometric automata. This is an instance of the transition from topological data (sets of points) to algebraic structures (lattices of open subsets) which is fundamental to locale theory, [6]. This result links our ideas on causality with Winskel's theory of events in computation, [13]; it provides a syntax for event structures and it opens the way to giving a causal interpretation of event structure phenomena. We also show that geometric automata give rise to domains of configurations which are more general than the event domains of Winskel and Droste, [2].

In the concluding section we discuss briefly some open problems and some interesting directions for further study.

2 Requirements for A Logic of Causality

In this section we discuss three general principles for a logic of causality. These principles encapsulate some basic intuitions about events and causality. They should not be regarded as cast in stone, but rather as defining the parameters of the present investigation.

2.1 The principle of events

This principle of events asserts that causality exists at the level of action occurrences (events) and not at the level of actions. As I type this line of text at my word-processor I perform various actions. For instance, the action of pressing the "a" key. Each time this action occurs, a separate event is generated (a new ASCII character is added to my text file). I could, if I was sufficiently painstaking, enumerate these events from the very first time that I pressed the "a" key during this session of typing. Each time the action occurs, the reasons for it and the context in which it happens are different. The principle of events merely points out that in describing causal relationships we must do so at event level and not at action level because different action occurrences may have different causes.

All this may seem rather obvious. However, it means that when we look at a reactive system and attempt to describe it from this causal viewpoint we have to be able to see the individual events. The system must be transparent and not a black box. If we observe the system in operation in this way we will necessarily see a sequence of distinct events.

Definition 2.1 *A string $s \in E^*$ over some alphabet E is said to be pure if each symbol in E appears at most once in s.*

We will use the notation E^{*p} for the set of pure strings over E. Recall that a subset $T \subseteq E^*$ is a trace set if T is non-empty and prefix-closed: $\varepsilon \in T$ and $st \in T \Rightarrow s \in T$. Note that E^{*p} is itself a trace set. A pure trace set is a trace set contained in E^{*p}.

Transparent systems generate pure trace sets. Not only do we regard each symbol on a trace as a separate event but we regard the same symbol on different traces as the same event. Hence the trace set $\{\varepsilon, a, b, ab, ba\}$ can be unambiguously reconstructed into the reactive system which has two events a and b which are concurrent. There are other reactive systems which might appear to generate the same set of traces, for instance one where there is a choice between the two interleavings of a and b. But transparently, this system has four events, although only two actions. We would actually see traces like $\{\varepsilon, u, v, uw, vx\}$ where, say, events u and x are occurrences of action a and events v and w are occurrences of b.

It appears from this discussion that pure trace sets are equivalent to transparent reactive systems and we might as well throw away the systems and work with the trace sets. If our aim is to set up a formalism based on a logic for causality then the power of the logic - its expressibility - should be assessed by determining the class of pure trace sets which the formalism generates. This is the mathematical content of the principle of events.

Actions can be recovered from their underlying events through a labelling function as is done for labelled event structures and pomsets. This amounts to a process of abstraction: different events are deemed to represent the same action. In this paper we shall not be concerned with labelled automata.

2.2 The principle of irrelevance of history

Consider the pure trace set $T_1 = \{\varepsilon, a, b, ab, ba, abc, bad\}$. The events c and d depend on the order in which the events a and b have occurred. If a happens before b then c occurs but not d and vice-versa. In other words, history is relevant. To describe this in our logic of causality it seems inescapable that we would need logical formulae $f \in \mathcal{L}(E)$ whose truth depends on sequences of events. This would appear to be a more discriminating and complex logic than one based on sets of events. In this paper we choose to avoid this possibility; for us, history will not be relevant. We stress that this is purely for reasons of convenience. We believe that history is very relevant and that we must come to terms with it in the future if we are to fully understand causality. By treating first the simpler problem we hope to lay the foundation for a study of the more complex one. We shall mention some of the directions that might be taken in §5.

The principle of irrelevance of history has two mathematical consequences. Firstly, it tells us something about the model theory of our logic. The truth of a logical formula will be determined by sets, as opposed to sequences, of events. Hence we expect a pairing between sets of events, $S \in 2^E$, and formulae, $f \in \mathcal{L}(E)$:

$$S(f) = \begin{cases} \mathsf{T} \\ \mathsf{F}. \end{cases}$$

Secondly, it rules out of consideration certain pure trace sets, such as the one above, in assessing our logic. How can we describe the trace sets for which history is irrelevant?

We need some notation for strings to start with. The prefix ordering on strings, $s, t \in E^*$, will be denoted $s \leq t$. The notation $s \sim t$ will indicate that s and t are permutations of each

other. If T is a trace set and $s \in T$, then $[s]$ will indicate the equivalence class of s up to permutation in T:

$$[s] = \{t \in T | t \sim s\}.$$

On such equivalence classes, which form the set $T/\sim$, one can define the prefix ordering up to permutation: $[s] \preceq [t]$ if, and only if, $\exists s_1 \in [s], t_1 \in [t]$ such that $s_1 \leq t_1$. It is easy to see that this relation on $T/\sim$ is reflexive and anti-symmetric. It is not, however, always transitive, as the trace set T_1 above makes clear: $[a] \preceq [ab]$ and $[ab] \preceq [bad]$ but $[a] \not\preceq [bad]$. This captures exactly what we mean by history-relevance.

Definition 2.2 *A trace set T is transitive if the prefix ordering up to permutation is transitive. That is, if $(T/\sim, \preceq)$ is a partial order.*

For the rest of this paper we shall concentrate on pure, transitive trace sets. The partial order $(T/\sim, \preceq)$ will turn out to be particularly significant when we consider domains of configurations.

2.3 The principle of finite causes

It is clear that our logic of causality must have a connective resembling AND. When there are infinitely many events available this raises the possibility of an infinitary conjunction:

$$e \to \bigwedge \{e_1, e_2, \cdots, e_n, \cdots\}.$$

The problem with this is clear: the event e could not take place until infinitely many events had occurred. An infinitary disjunction, on the other hand, is much better behaved:

$$e \to \bigvee \{e_1, e_2, \cdots, e_n, \cdots\}.$$

Here e could occur after any of the e_i, a form of infinite choice. It is interesting that the intuitive duality between AND and OR which we argued for in §1 disappears in an infinite context.

The principle of finite causes is customarily invoked - in event structures for instance - as a way of avoiding infinitary conjunction. We could, of course, simply rule out the use of infinitary conjunction in our logic of causality. However it is more insightful to state the principle of finite causes in terms of the model theory in the following way: the validity of any formula in our logic should be determined by its value on just the finite sets of events. More precisely, if $Fin(E) \subseteq 2^E$ denotes the set of finite subsets of E, and $f, g \in \mathcal{L}(E)$ are formulae in our logic such that

$$\{S \in Fin(E) | S(f) = \mathsf{T}\} \;=\; \{T \in Fin(E) | T(g) = \mathsf{T}\}$$

then $f = g$ in $\mathcal{L}(E)$. This has the effect of directing our attention towards the set $Fin(E)$. Its structure seems to be of great significance in understanding logics of causality.

This completes our discussion of the principles required by a logic of causality. In the next section we introduce the particular candidate, geometric logic, which we will study in the rest of the paper.

3 Geometric Automata

The principle of finite causes draws our attention to the discrepancy between infinitary OR and infinitary AND. This should warn us that the customary dualities of classical Boolean logic are threatened in the infinite case. A similar phenomenon has been observed in geometric logic, the "logic of finite observations", [12, Chapter 2], for much the same reasons. This suggests that we examine geometric logic as a candidate for a logic of causality.

Geometric logic is an infinitary intuitionistic logic. The Lindenbaum algebra of a geometric theory is referred to as a frame, [6, 12]. Recall that if $(F, \leq)$ is a poset then a subset $S \subseteq F$ has a meet (greatest lower bound), $m = \bigwedge S$, if $m \leq x$ for all $x \in S$ (m is a lower bound) and, if m' has the same property, then $m' \leq m$ (m is the greatest lower bound). A similar definition holds for the join (least upper bound) of S, denoted $\bigvee S$. Note that $x \leq y$ if, and only if, $x \wedge y = x$; similarly, $x \leq y$ if, and only if, $x \vee y = y$.

Definition 3.1 *A frame F is a poset in which* (1) *all finite meets exist;* (2) *arbitrary joins exist;* (3) *binary meets distribute over arbitrary joins, $a \wedge \bigvee_{i \in I}\{b_i\} = \bigvee_{i \in I}\{a \wedge b_i\}$.*

Historically, frames arose not from logic but from topology: they were the posets of open subsets of a topological space, In a topological space, meet corresponds to intersection of open subsets and join to union; the distributive law comes for free. The move away from topology as "sets of points" to topology as "lattices of opens" (pointless topology or locale theory) is ably documented in Johnstone's survey, [6].

We denote $\bigwedge \emptyset$, the greatest element of F, by T, and $\bigvee \emptyset$, the least element, by F. The simplest non-degenerate frame, the Sierpinski frame, is the poset $2 = \{\mathsf{F}, \mathsf{T}\}$ where, of course, $\mathsf{F} \leq \mathsf{T}$. If F and G are frames, a frame homomorphism is a function $f : F \to G$ which preserves meets and joins and hence preserves the partial order and T and F. Frames and frame homomorphisms form a category, **Frm**. Taking the poset of open subsets is a (contravariant) functor from the category, **Top**, of topological spaces and continuous maps, to **Frm**. (In pointless topology one prefers to work in $\mathbf{Frm}^{op}$, the category of locales, [6], but this distinction need not concern us here.)

When compared with complete Boolean algebras and complete Heyting algebras, frames have a better algebraic theory. Despite the infinitary operation, they can still be constructed by generators and relations. (**Frm** is algebraic over **Set**, [5, §II.1.2].) Given a set E there exists a free frame generated by E, denoted $Fr(E)$. $Fr : \mathbf{Set} \to \mathbf{Frm}$ is the left adjoint to the forgetful functor and $Fr(E)$ is defined up to isomorphism by the following universal property. There exists a set function $i : E \to Fr(E)$ which is initial in the following sense: given any other set function $f : E \to G$ from E to some frame G, there exists a unique frame homomorphism $\overline{f} : Fr(E) \to G$ such that $f = \overline{f}i$. We shall drop the function i and identify elements of E with their images $i(e) \in Fr(E)$. Similarly, we shall use the same notation for the set function f and its lifting to a frame homomorphism.

In marked contrast to frames there is no free complete Boolean algebra or free complete Heyting algebra on countably infinitely many generators, [5, §I.4.10], a problem which will confront us later. For the moment our first concern is with the structure of $Fr(E)$.

If $(X, \leq)$ is a poset, $x\uparrow$ will denote the upward closure of $x \in X$: $x\uparrow = \{y \in X \mid x \leq y\}$. This can be extended to subsets: if $S \subseteq X$, then $S\uparrow = \bigcup_{x \in S}\{x\uparrow\}$. A subset S is upward closed if $S = S\uparrow$. Downward closures will be denoted $x\downarrow$, $S\downarrow$. The collection of upward (downward) closed subsets of $(X, \leq)$ forms a topology on X, the Alexandrov topology, [12, §3.6.2], denoted $X\uparrow$ ($X\downarrow$).

Of particular interest to us are topologies on $Fin(E)$ which we consider as a partial order under inclusion of subsets. Let $\theta : E \to Fin(E)\uparrow$ be defined by $\theta(e) = \{e\}\uparrow$. As discussed above, this lifts uniquely to a frame homomorphism on $Fr(E)$.

Proposition 3.1 *$\theta : Fr(E) \to Fin(E)\uparrow$ is an isomorphism of frames.*

Proof: Johnstone's proof that **Frm** is algebraic over **Set**, [5, §II.1.2], shows that $Fr(E)$ may be represented, up to isomorphism, as $MSL(E)\downarrow$ where $MSL(E)$ is the free meet semi-lattice generated by E. In this representation, a generator $e \in E$ corresponds to $e\downarrow$ in $MSL(E)\downarrow$. Now $MSL(E)$ is simply $Fin(E)^{op}$ as a poset and hence $MSL(E)\downarrow$ can be identified with $Fin(E)\uparrow$. Under this identification, the generator e corresponds to $\{e\}\uparrow$. The result follows.

QED

The real content of this result is, of course, Johnstone's construction of the free frame. However, the result has a special significance for us: it is the "crucial characteristic property of $\{\mathsf{AND}, \mathsf{OR}\}$ causality", [3, Lemma 3.1], placed in its correct setting and it is the key to our causal interpretation of event structures in §4.

In order to relate frames to the discussion in §2 we need to reintroduce the logical dimension which has been missing so far. In the trivial frame, $\mathsf{2}$, the operations of meet and join are identical to AND and OR if T and F are interpreted as "true" and "false". Let E be a set and $v : E \to \mathsf{2}$ be any set function. By the universal property of free frames, v lifts to a frame homomorphism $v : Fr(E) \to \mathsf{2}$. This corresponds to a valuation on $Fr(E)$ in which the elements $e \in E$ with $v(e) = \mathsf{T}$ are given the value "true", all other elements of E are valued "false" and meet and join are interpreted as AND and OR. We shall identify a function $v : E \to \mathsf{2}$ with the characteristic function of the subset $\{e \in E \mid v(E) = \mathsf{T}\}$ so that statements like $v \subseteq w$ have an obvious meaning. If $S \subseteq E$ then v_S will denote the corresponding valuation. We have a pairing

$$v(f) = \begin{cases} \mathsf{T} \\ \mathsf{F} \end{cases}$$

between subsets $v \in 2^E$ and formulae $f \in Fr(E)$. This is what we expect from the principle of irrelevance of history as discussed in §2.2.

Lemma 3.1 *For any $f \in Fr(E)$, $\theta(f) = \{v \in Fin(E) \mid v(f) = \mathsf{T}\}$.*

Proof: Let $\theta_1 : Fr(E) \to Fin(E)\uparrow$ be defined by $\theta_1(f) = \{v \in Fin(E) \mid v(f) = \mathsf{T}\}$. Since the frame operations in $Fin(E)\uparrow$ are set theoretic, it is easy to check that $\theta_1(f \wedge g) = \theta_1(f) \cap \theta_1(g)$ and $\theta_1(\bigvee_{i\in I}\{f_i\}) = \bigcup_{i\in I}\{\theta_1(f_i)\}$. Hence θ_1 is a homomorphism of frames. By the universal property of $Fr(E)$ it is sufficient to check that $\theta = \theta_1$ on elements of E. For $e \in E$, and any valuation $v : E \to \mathsf{2}$, $v(e) = \mathsf{T}$ if, and only if, $e \in v$. That is, if, and only if, $v \in \{e\}\uparrow$. It follows that $\theta(e) = \{e\}\uparrow = \{v \in Fin(E) \mid v(e) = \mathsf{T}\} = \theta_1(e)$. Hence $\theta = \theta_1$ and the result follows.

QED

The Lemma shows immediately that the principle of finite causes, §2.3, is obeyed. It appears that $Fr(E)$ is a good candidate for $\mathcal{L}(E)$. However, as it stands, a table of the form $\rho : E \to Fr(E)$ would be incapable of representing the trace set $\{\varepsilon, a, b\}$. Indeed, when E is finite, such tables are exactly the $\{\mathsf{AND}, \mathsf{OR}\}$ automata of [3] which represent only the confluent trace sets, [3, Proposition 3.1]. This brings us to the major problem of a causal logic: dealing with choice or conflict. (We will return to the question of expressibility of the logic later in this section.)

It seems clear that conflict is related logically to negation. If we attempt to incorporate negation into geometric logic we move inexorably towards a full Boolean algebra. Of course, frames have a negation anyway, the pseudocomplement of the corresponding complete Heyting algebra, [5, §I.1.11], [12, §3.10]. As one might expect, this has quite the wrong interpretation in the model theory: the pseudocomplement of an upwards closed subset is always empty. The correct interpretation comes from looking at both the upward and downward closed subsets of $Fin(E)$, corresponding to formulae in $Fr(E)$ and their "negations". The smallest topology containing both $Fin(E)\uparrow$ and $Fin(E)\downarrow$ is evidently the discrete topology on $Fin(E)$: the complete Boolean algebra $2^{Fin(E)}$. In the finite case this is free; in the infinite case we know that it cannot be, as pointed out earlier. It is not at all clear how $2^{Fin(E)}$ can be presented algebraically.

In this paper we shall sidestep this problem by dealing with choice in the operational semantics of the table formalism rather than within the logic of causality. This trick is entirely contrary to the spirit of the present paper but is justified because of the insight it gives into event structures. Because of the general acceptance of event structures as a formalism for dealing with concurrency, [11, 10, 14], and their importance as a link to the domain theory of Scott, [2, 8, 13], we feel that a causal interpretation of them is interesting enough to defer a purely logical treatment of conflict.

Definition 3.2 *A geometric automaton, G, is a triple, (E, ρ, σ) where E is a set of events and $\rho, \sigma : E \to Fr(E)$ are a pair of functions from E to the free frame generated by E.*

We will sometimes use subscripts, E_G, ρ_G, σ_G to avoid confusion. E is the carrier of G and ρ and σ are, respectively, the positive and negative causality functions. Conflict arises from the tension between them as the following operational semantics makes clear. Recall that for any set E, $v_\emptyset$ is the all-false valuation: $v_\emptyset(e) = \mathsf{F}$ for all $e \in E$.

Definition 3.3 *Let $G = (E, \rho, \sigma)$ be a geometric automaton. The event $e \in E$ is said to be enabled in G if $v_\emptyset(\rho(e)) = \mathsf{T}$ and $v_\emptyset(\sigma(e)) = \mathsf{F}$.*

Note that, by Proposition 3.1 and Lemma 3.1, $v_\emptyset(f) = \mathsf{T}$ if, and only if, $f = \mathsf{T}$.

Definition 3.4 *The automaton $G = (E, \rho, \sigma)$ offers the event e and evolves into the automaton $G' = (E', \rho', \sigma')$, denoted in the usual way by $G \xrightarrow{e} G'$, if, and only if, the following conditions hold:* (1) *e is enabled in G;* (2) *$E' = E - \{e\}$;* (3) *$\rho' = \rho[\mathsf{T}/e]$;* (4) *$\sigma' = \sigma[\mathsf{T}/e]$.*

The use of $v_\emptyset$ corresponds to setting each of the generators in E to F; informally, no event has yet been offered. Those events which can be offered are the enabled ones: those for which the positive causality is "on", T, and the negative causality is "off", F. When e is offered, we substitute for this generator the value T, as in condition 3 above, and a new automaton emerges.

An example may help to make the semantics less abstract. Consider the automaton G shown below, where the positive causality is written in the first column of the table and the negative causality in the last column.

$$G = \boxed{\begin{array}{ccc} \mathsf{T} & a & \mathsf{F} \\ \mathsf{T} & b & \mathsf{F} \\ a \vee b & c & a \wedge b \end{array}}.$$

According to Definition 3.3, the only enabled events are a and b. If a is offered, the automaton evolves into

$$H = \boxed{\begin{array}{ccc} \mathsf{T} & b & \mathsf{F} \\ \mathsf{T} & c & b \end{array}}$$

since $\mathsf{T} \vee b = \mathsf{T}$ and $\mathsf{T} \wedge b = b$. The event c has now become enabled. If c is offered, then we evolve to the automaton

$$K = \boxed{\begin{array}{ccc} \mathsf{T} & b & \mathsf{F} \end{array}}$$

which can then proceed to offer b. On the other hand, if H had offered b instead of c, we would evolve to the automaton

$$L = \boxed{\begin{array}{ccc} \mathsf{T} & c & \mathsf{T} \end{array}}$$

in which c is no longer enabled. The automaton L is dead and can offer no events. The behaviour of G if it offers b to begin with is symmetrical. The traces of G are seen to be $\{\varepsilon, a, b, ac, bc, ab, ba, acb, bca\}$.

An automaton H is said to be derived from G if there are a sequence of automata $G_1, \cdots, G_n$, where $n \geq 1$, such that,

$$G = G_1 \stackrel{e_1}{\rightarrow} G_2 \stackrel{e_2}{\rightarrow} \cdots \stackrel{e_{n-1}}{\rightarrow} G_n = H.$$

The set of derived automata of G is denoted $Der(G)$. Note that $G \in Der(G)$, corresponding to the case $n = 1$, when $G = H$. $Der(G)$ forms a labelled transition system, $\mathsf{LTS}(G)$, under the operational semantics defined above:

$$\mathsf{LTS}(G) = (Der(G), E_G, \stackrel{e}{\rightarrow}),$$

where $\stackrel{e}{\rightarrow} \subseteq Der(G) \times E_G \times Der(G)$. The string $e_1 e_2 \cdots e_{n-1} \in (E_G)^{*p}$ is referred to as a trace of G. The empty string, ε, corresponds to the case $n = 1$. The set of traces is denoted $traces(G)$; it is clearly a pure trace set as defined in §2.1. Since we have carefully avoided history relevance, the following result is hardly surprising.

Lemma 3.2 *If G is a geometric automaton then $traces(G)$ is a pure transitive trace set.*

However, not all such trace sets arise in this way. It is not hard to show that the following pure transitive trace set $\{\varepsilon, a, c, ab, cd, abc, cda, abcd, cdab\}$ cannot be the traces of any geometric automaton. Hence geometric logic, despite the trick of negative causality, is not as expressive of causal behaviour as we would ideally like. In this respect we mention without proof that any finite pure transitive trace set is the traces of some causal automaton (ie: where full Boolean logic is used). This draws our attention once again to problem of the structure of $2^{Fin(E)}$ when E is infinite.

An alternative discussion of the operational semantics of geometric automata may be found in [4]. It is equivalent to the one given above but recasts the behaviour in terms of the concepts of "observation" and "state". It is, in some ways, a more attractive treatment but it is unsuited to discussing event structures and we have chosen not to adopt it here.

This completes the introductory material on geometric automata. Before embarking on the next section it will be convenient to record a lemma which we will need later. Suppose that $e \in E$ and $F = E - \{e\}$. If $c \subseteq Fin(E)$ let $r_e(c)$ be defined by

$$r_e(c) = \{S \in Fin(F) \mid \text{either } S \in c, \text{ or, } S \cup \{e\} \in c\}.$$

It is easy to check that if c is upward closed then so is $r_e(c)$ so that we have a function $r_e : Fin(E)\uparrow \rightarrow Fin(F)\uparrow$. Note that r_e also preserves downward closed subsets.

Lemma 3.3 *r_e is a homomorphism of frames. Furthermore, the diagram below commutes*

$$\begin{array}{ccc} Fin(E)\uparrow & \xleftarrow{\theta_E} & Fr(E) \\ \downarrow r_e & & \downarrow [\mathsf{T}/e] \\ Fin(F)\uparrow & \xleftarrow{\theta_F} & Fr(F). \end{array}$$

Proof: The frame operations in $Fin(E)\uparrow$ and $Fin(F)\uparrow$ are set theoretic and it is easy to check that $r_e(c \cap d) = r_e(c) \cap r_e(d)$ and $r_e(\bigcup_{i\in I}\{c_i\}) = \bigcup_{i\in I}\{r_e(c_i)\}$, where $c, d, c_i \in Fin(E)\uparrow$.

The function $[\mathsf{T}/e] : Fr(E) \to Fr(F)$ is also a homomorphism of frames: it is the lifting to $Fr(E)$ of the function $E \to Fr(F)$ which is the identity on $x \neq e$ and takes e to T. Hence it is sufficient, by the universal property of $Fr(E)$, to check the commutativity of the diagram on the elements $x \in E$. If $x \neq e$ then $\theta_F(x[\mathsf{T}/e]) = \theta_F(x) = \{x\}\uparrow^F = r_e(\{x\}\uparrow^E) = r_e(\theta_E(x))$. (We have used superscripts to indicate in which set the upward closure is taking place.) If $x = e$, then $\theta_F(e[\mathsf{T}/e]) = \theta_F(\mathsf{T}) = \emptyset\uparrow^F = r_e(\{e\}\uparrow^E) = r_e(\theta_E(e))$.

QED

4 Event Structures

In this section we show how an event structure can be interpreted as a special type of geometric automaton. We recall the definition of an event structure from [14, §1.1.1].

Definition 4.1 *An event structure, S, is a triple $S = (E, Con, \vdash)$ where (1) E is a set of events; (2) $Con \subseteq Fin(E)$, the consistency predicate, satisfies: if $v \in Con$ and $w \subseteq v$ then $w \in Con$; (3) $\vdash \subseteq Fin(E) \times E$, the enabling relation, satisfies: if $v \vdash e$ and $v \subseteq w$ then $w \vdash e$.*

(We use subscripts, E_S, Con_S, $\vdash_S$, where necessary, to avoid confusion.)

We have made one alteration to Winskel's original definition. The enabling relation, $\vdash$, is taken to lie in $Fin(E) \times E$ rather than $Con \times E$. In effect, if $v \in Con$ and $v \vdash e$, then we adjoin to the enabling relation in $Con \times E$ any $w \in Fin(E)$, such that $v \subseteq w$. This cannot create any new enablings of e from among the consistent subsets because of 4.1(3). The reader can easily check that this change makes no difference to the configurations of the event structure. Note that 4.1(2) merely states that Con is downward closed and 4.1(3), because of the change made above, states that $\{v \in Fin(E) \mid v \vdash e\}$ is upward closed for all $e \in E$.

Winskel's definition of the behaviour of an event structure is in terms of its configurations, [14, §1.1.2]. The configurations form a Scott domain, $\mathcal{D}(E)$, under inclusion of subsets. The behaviour of a geometric automaton, however, is expressed in terms of a labelled transition system. We need common ground in order to compare the two formalisms and so we present a transition system semantics for event structures. We then explain how the domain of configurations can be recovered from it.

Definition 4.2 *Let $S = (E, Con, \vdash)$ be an event structure. The event $e \in E$ is said to be enabled in S if $\emptyset \vdash e$ and $\{e\} \in Con$.*

Definition 4.3 *The event structure $S = (E, Con, \vdash)$ offers the event e and evolves into the event structure $S' = (E', Con', \vdash')$, denoted $S \xrightarrow{e} S'$, if, and only if, the following conditions hold: (1) e is enabled in S; (2) $E' = E - \{e\}$; (3) $Con' = r_e(Con)$; (4) $\forall x \in E'$, $\{v \in Fin(E') \mid v \vdash' x\} = r_e(\{w \in Fin(E) \mid w \vdash x\})$.*

Since r_e preserves upward or downward closed subsets of $Fin(E)$, as discussed in §3, this definition does indeed yield an event structure. In a similar way to geometric automata, we get a set of derived event structures, $Der(S)$, a labelled transition system, $\mathsf{LTS}(S) = (Der(S), E_S, \xrightarrow{e})$, and a set of traces, $traces(S)$.

Lemma 4.1 *If S is an event structure then $traces(S)$ is a pure transitive trace set.*

If S is an event structure and $s \in traces(S)$ then s gives rise to a (finite) subset of E_S, $[s]$, by simply forgetting the ordering of symbols in the trace. If $s, t \in traces(S)$ are permutations of each other then clearly $[s] = [t]$. Hence $[-]$ is a function from $traces(S)/\!\sim$ to $Fin(E_S)$. The configurations of an event structure are also subsets of E_S. Let $\mathcal{D}_0(E_S)$ denote the partial order of finite (compact) elements, [14, §1.1.15], of $\mathcal{D}(E_S)$. We know that the finite elements are exactly the finite configurations, [14, Theorem 1.1.16], and so $\mathcal{D}_0(E_S) \subseteq Fin(E_S)$ and inherits the partial order coming from inclusion of subsets.

Proposition 4.1 *If S is an event structure then $[-]$ induces an isomorphism of partial orders*

$$[-] : (traces(S)/\!\sim, \preceq) \to \mathcal{D}_0(E_S).$$

This makes it clear how the labelled transition system semantics defined above relates to Winskel's semantics in terms of configurations. Since $\mathcal{D}(E_S)$ is an algebraic directed complete partial order, [14, Theorem 1.1.6], we can recover the full domain from its finite elements by ideal completion, [12, Proposition 9.1.4]. This construction of the domain of configurations appears to be new although it seems implicit in [10, Theorem 5.1]. It has little to recommend it, in so far as event structures are concerned, because it is unnecessarily complicated and fails to give the infinite configurations. Its proper significance is only revealed in the context of geometric automata.

Definition 4.4 *If $S = (E, Con, \vdash)$ is an event structure, define the geometric automaton $\Delta(S) = G$ where:* (1) $E_G = E$; (2) $\rho_G(e) = \theta^{-1}(\{v \in Fin(E) \mid v \vdash e\})$; (3) $\sigma_G(e) = (\theta^{-1}(\overline{Con}))[\mathsf{T}/e]$.

(Here, $\overline{Con}$ denotes the complement of Con in $Fin(E)$. Evidently, $\overline{Con}$ is upwards closed.)

Event structures, as originally formulated by Winskel, have a topological character. Geometric automata are a reformulation in algebraic terms in keeping with the transition from topological data (sets of points) to algebraic structures (lattices of open subsets) which is fundamental to locale theory, [6]. Proposition 3.1 is the key to making this work as the definition above shows. The trick of using positive and negative causality for geometric automata is already implicit in the definition of event structures. Of course, we still have to show that the behaviour of an event structure S is the same as that of $\Delta(S)$. This is accomplished in the following lemmas.

Lemma 4.2 *If S is an event structure then e is enabled in S if, and only if, e is enabled in $\Delta(S)$.*

Proof: Let $S = (E, Con, \vdash)$ and $\Delta(S) = (E, \rho, \sigma)$ and abbreviate "if, and only if," to iff. By Definitions 4.4(2) and 4.1(3), $\emptyset \vdash e$ iff $\theta(\rho(e)) = \emptyset\!\uparrow$. That is, by Lemma 3.1, iff $v_\emptyset(\rho(e)) = \mathsf{T}$.

Now let $F = E - \{e\}$. By Lemma 3.3 $(\theta_E^{-1}(\overline{Con}))[\mathsf{T}/e] = \theta_F^{-1}(r_e(\overline{Con}))$. Now $\{e\} \in Con$ iff $\{e\} \notin \overline{Con}$ iff $r_e(\overline{Con}) \neq \emptyset\uparrow$. Hence, by Definition 4.4(3) and Lemma 3.1, $\{e\} \in Con$ iff $v_\emptyset(\sigma(e)) = \mathsf{F}$.

The result now follows from Definitions 3.3 and 4.2.

QED

Lemma 4.3 *If S is an event structure and $S \xrightarrow{e} T$ then $\Delta(S) \xrightarrow{e} \Delta(T)$. Conversely, if G is a geometric automaton of the form $G = \Delta(S)$ and $G \xrightarrow{e} H$, then $H = \Delta(T)$ and $S \xrightarrow{e} T$.*

We can now put together the main result of this paper.

Theorem 4.1 *Δ gives a bijective correspondence between event structures and geometric automata, G, with the property that $\exists f \in Fr(E_G)$ such that $\forall e \in E_G$, $\sigma_G(e) = f[\mathsf{T}/e]$. Furthermore, for an event structure $S = (E, Con, \vdash)$, Δ induces an isomorphism of transition systems labelled over E between $\mathsf{LTS}(S)$ and $\mathsf{LTS}(\Delta(S))$.*

Proof: If $G = (E, \rho, \sigma)$ is an automaton of the appropriate form, let $\Lambda(G) = (E, Con, \vdash)$ be defined by $Con = \overline{\theta(f)}$, where $f \in Fr(E)$ is the element in the statement of the theorem, and $\vdash = \{(v, e) \in Fin(E) \times E \mid v \in \theta(\rho(e))\}$. It now follows from Proposition 3.1 that $\Delta(\Lambda(G)) = G$ and, for an event structure S, that $\Lambda(\Delta(S)) = S$. This proves the first part. The second part follows easily from Lemma 4.3.

QED

The result shows that event structures are a special case of geometric automata: those with "constant" negative causality. With this in place we now have an obvious way to generalise Winskel's domain of configurations. If X is a poset, $Idl(X)$ denotes its ideal completion in the sense of [12, Definition 9.1.1].

Definition 4.5 *If G is a geometric automaton, its domain of configurations, denoted $\wp(G)$, is defined by $\wp(G) = Idl(traces(G)/\sim, \preceq)$.*

If follows from Proposition 4.1 that for an event structure S, $\wp(\Delta(S)) \cong \mathcal{D}(S)$. In an event structure the partial order on configurations comes from inclusion of subsets. For geometric automata this is no longer the case: a configuration may be properly contained in another one but the automaton may not be able to evolve from the smaller state to the larger. The inclusion ordering does not reflect the behaviour of the automaton. The correct partial order is that inherited from the prefix ordering on traces. For example, the automaton G discussed in §3 has the domain shown in Figure 3. We see that the configuration $\{a, b\}$ cannot evolve into $\{a, b, c\}$.

This example is also interesting because the domain of configurations is not consistently complete: the configurations $\{a\}$ and $\{b\}$ have an upper bound but no least upper bound. This shows that geometric automata give rise to more general domains than those coming from event structures, which Droste refers to as event domains, [2].

5 Conclusion

The main thrust of this paper has been the importance of studying causality from a logical standpoint. We have shown that the (first) crucial problem in this study is the logical treatment of conflict or choice. The solution adopted here enables us to give a simple interpretation of

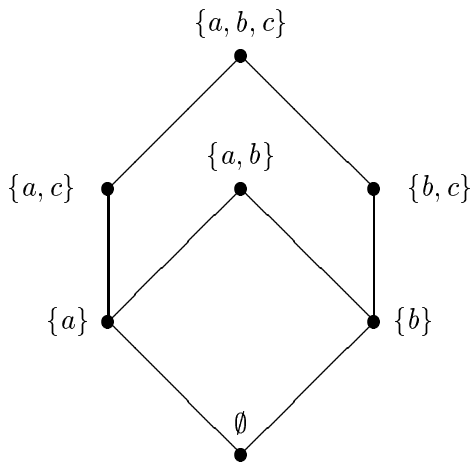

Figure 3: Domain of configurations

Winskel's general event structures but it cannot be considered as a complete and satisfactory answer in general.

The other interesting direction for further work is to drop the assumption of irrelevance of history. In this respect, our use of locales suggests that quantales would be the appropriate algebraic framework for history sensitive causality. The corresponding logic is (geometric) linear logic. Perhaps Girard's deconstruction of negation may also provide some clues towards a satisfactory logical treatment of conflict. We hope to address some of these speculations in future papers.

Acknowledgements

The work outlined here was undertaken as part of project CERES at Hewlett-Packard's Research Laboratory in Bristol, England. I am grateful to an anonymous ICALP referee of an earlier paper for pointing out the importance of infinitary OR. Martin Hyland raised the possibility of using logics other than Boolean logic. These clues were the starting point of the present study. Peter Johnstone kindly rescued me from my own stupidity and set me straight on the proof of Proposition 3.1, for which I am most grateful. My thanks also to the referees of the present paper for pointing out some errors and for their comments on presentation.

References

[1] S. Abramsky. Domain theory in logical form. *Annals of Pure and Applied Logic*, 1989.

[2] M. Droste. Event structures and domains. *Theoretical Computer Science*, 68:37–47, 1989.

[3] J. Gunawardena. Causal Automata I: Confluence $\equiv \{AND, OR\}$-Causality. In M. Z. Kwiatkowska, M. W. Shields, and R. M. Thomas, editors, *Semantics for Concurrency*, pages 137–156. Springer Workshops in Computing, 1990. To appear in TCS.

[4] J. Gunawardena. Eventless structures? (extended abstract). In *Third Workshop on Concurrency and Compositionality*, Goslar, March 1991.

[5] P. T. Johnstone. *Stone Spaces*, volume 3 of *Studies in Advanced Mathematics*. Cambridge University Press, 1982.

[6] P. T. Johnstone. The point of pointless topology. *Bulletin American Mathematical Society*, 8(1):41–53, 1983.

[7] R. Milner. *Communication and Concurrency*. International Series in Computer Science. Prentice-Hall, 1989.

[8] M. Nielsen, G. Plotkin, and G. Winskel. Petri nets, event structures and domains. *Theoretical Computer Science*, 13:85–108, 1981.

[9] M. Smyth. Powerdomains and predicate transformers: a topological view. In J. Diaz, editor, *Automata, Languages and Programming*. Springer LNCS 154, 1983.

[10] P. S. Thiagarajan. Some behavioural aspects of net theory. In T. Lepistö and A. Salomaa, editors, *Automata, Languages and Programming*. Springer LNCS 317, 1988.

[11] R. van Glabbeek and U. Goltz. Equivalence notions for concurrent systems and refinement of actions. Arbeitspapiere 366, GMD, February 1989.

[12] S. Vickers. *Topology via Logic*, volume 5 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 1989.

[13] G. Winskel. *Events in Computation*. PhD thesis, University of Edinburgh, 1980.

[14] G. Winskel. Event structures. In W. Brauer, W. Reisig, and G. Rozenberg, editors, *Advances in Petri Nets*. Springer LNCS 255, 1987.