Minimisation of Event Structures

Paolo Baldan, Alessandra Raffaetà

[a] *University of Padova, Italy*
[b] *University Ca' Foscari of Venice, Italy*

Abstract

Event structures are fundamental models in concurrency theory, providing a representation of events in computation and of their relations, notably concurrency, conflict and causality. In this paper we present a theory of minimisation for event structures. Working in a class of event structures that generalises many stable event structure models in the literature (e.g., prime, asymmetric, flow and bundle event structures), we study a notion of behaviour-preserving quotient, referred to as a folding, taking (hereditary) history-preserving bisimilarity as a reference behavioural equivalence. We show that for any event structure a folding producing a uniquely determined minimal quotient always exists. We observe that each event structure can be seen as the folding of a prime event structure, and that all foldings between general event structures arise from foldings of (suitably defined) corresponding prime event structures. This gives a special relevance to foldings in the class of prime event structures, which are studied in detail. We identify folding conditions for prime and asymmetric event structures, and show that also prime event structures always admit a unique minimal quotient (while this is not the case for various other event structure models).

Keywords: Event structures, minimisation, history-preserving bisimilarity, behaviour preserving quotient

1. Introduction

When dealing with formal models of computational systems, a classical problem is that of minimisation, i.e., for a given system, define and possibly construct a compact version of the system which, very roughly speaking, exhibits the same behaviour as the original one, avoiding unnecessary duplications. The minimisation procedure depends on the notion of behaviour of interest and also on the expressive power of the formalism at hand, which determines its capability of describing succinctly some behaviour. One of the most classical examples is that of finite state automata, where one is typically interested in the accepted

*This work is supported by the Ministero dell'Universtà e della Ricerca Scientifica of Italy, under Grant No. 201784YSZ5, PRIN2017 – ASPRA.

language. Given a deterministic finite state automaton, a uniquely determined minimal automaton accepting the same language can be constructed, e.g., as a quotient of the original automaton via a partition/refinement algorithm (see, e.g., [1]). Moving to non-deterministic finite automata, minimal automata become smaller, at the price of a computationally more expensive minimisation procedure and non-uniqueness of the minimal automaton [2].

In this paper we study the problem of minimisation for event structures, a fundamental model in concurrency theory [3, 4, 5]. Event structures are a natural semantic model when one is interested in modelling the dynamics of a system by providing an explicit representation of the events in computations (occurrences of atomic actions) and of the relations between events, like causal dependencies, choices, possibility of parallel execution, i.e., in what is referred to as a true concurrent (non-interleaving) semantics. Prime event structures [3], probably the most widely used event structure model, capture dependencies between events in terms of causality and conflict. A number of variations of prime event structures have been introduced in the literature. In this paper we will deal with asymmetric event structures [6], which generalise prime event structures with an asymmetric form of conflict which allows one to model concurrent readings and precedences between actions, and flow [7, 8] and bundle [9] event structures, which add the possibility of directly modelling disjunctive causes. Event structures have been used for defining a concurrent semantics of several formalisms, like Petri nets [3], graph rewriting systems [10, 11, 12] and process calculi (see, e.g., [13, 7, 14, 15, 16, 17, 18]). Recent applications are in the field of weak memory models [19, 20, 21] and of process mining and differencing [22].

Behavioural equivalences, defined in a true concurrent setting, take into account not only the possibility of performing steps, but also the way in which such steps relate with each other. We will focus on hereditary history-preserving (hhp-)bisimilarity [23], the finest equivalence in the true concurrent spectrum in [24], which, via the concept of open map, has been shown to arise as a canonical behavioural equivalence when considering partially ordered computations as observations [25].

The motivation for the present paper originally stems from some work on the analysis and comparison of business process models. The idea, advocated in [22, 26], is to use event structures as a foundation for representing, analysing and comparing process models. The processes, in their graphical presentation, should be understandable, as much as possible, by a human user, who should be able, e.g., to interpret the differences between two processes diagnosed by a comparison tool. For this aim it can be important to avoid "redundancies" in the representation and thus to reduce the number of events, but clearly without altering the behaviour. The paper [27] explores the use of asymmetric and flow event structures and, for such models, it introduces some ad hoc reduction techniques that allow one to merge sets of events without changing the true concurrent behaviour. A general notion of behaviour preserving quotient, referred to as a folding, is introduced over an abstract class of event structures, having asymmetric and flow event structures as subclasses. However, no general theory is developed. The paper focuses on a special class of foldings, the so-called elementary foldings, which can only merge a single set of events into one event, and these are studied separately on each specific subclass of event structures (asymmetric and flow event structures), providing only sufficient conditions ensuring that a function is a folding.

A general theory of behaviour preserving quotients for event structures is thus called for, settling some natural foundational questions. Is the notion of folding adequate, i.e., are all behaviour preserving quotients expressible in terms of foldings? Is there a minimal quotient in some suitably defined general class of event structures? What happens in specific subclasses? (For asymmetric and flow event structures the answer is known to be negative, but for prime event structures the question is open.) Working in the specific subclasses of event structures, can we have a characterisation of general foldings, providing not only sufficient but also necessary conditions?

In this paper we start addressing the above questions. We work in a general class of event structures based on the idea of family of posets in [28], sufficiently expressive to generalise most stable event structures models in the literature, including prime [3], asymmetric [6], flow [7] and bundle [9] event structures.

As a first step we study, in this general setting, the notion of folding, i.e., of behaviour preserving quotient. A folding is a surjective function that identifies some events while keeping the behaviour unchanged. Formally, it establishes a hhp-bisimilarity between the source and target event structure. Foldings can be characterised as open maps in the sense of [25]. Actually, it turns out that not all behaviour preserving quotients arise as a folding, but we show that for any behaviour preserving quotient, there is a folding that induces a coarser equivalence, in a way that foldings properly capture all possible behaviour preserving quotients. Additionally, given two possible foldings of an event structure we show that it is always possible to "join" them. This allows us to prove that for each event structure a maximally folded version, namely a uniquely determined minimal quotient always exists.

Relying on the order-theoretic properties of the set of configurations of event structures [28], and on the correspondence between prime event structures and domains [3], we derive that each event structure in the considered class arises as the folding of a canonical prime event structure. Moreover, all foldings between general event structures arise from foldings of the corresponding canonical prime event structures. Interestingly, this result can be derived from the characterisation of folding morphisms as open maps.

The results above give a special relevance to foldings in the class of prime event structures, which thus are studied in detail. We provide necessary and sufficient conditions characterising foldings for prime event structures. This characterisation of foldings can guide, at least in the case of finite structures, the construction of behaviour preserving quotients. Moreover we show that also prime event structures always admit a minimal quotient.

Relying on the characterisation of foldings we can also establish a clear connection with the so-called abstraction homomorphisms, introduced in [29] for similar purposes in a more restricted context.

The fact that all event structures arise as foldings of prime event structures allows one to think of various brands of event structures in the literature, like asymmetric, flow and bundle event structures, as more expressive models that allow for smaller realisations of a given behaviour, i.e., of smaller quotients. For all these classes, however, the uniqueness of the minimal quotient is lost. Despite the fact that foldings on wider classes of event structures can be studied on the corresponding canonical prime event structures, a direct approach can be theoretically interesting and it can lead to more efficient minimisation procedures. In this paper, a characterisation of foldings is explicitly devised for asymmetric event structures.

Most results have a natural categorical interpretation. In order to keep the presentation simple, the categorical references are inserted in side remarks (and sometimes used in proofs) that can be safely skipped by the non-interested reader. This applies, in particular, to the possibility of viewing foldings as open maps in the sense of [25], which is discussed in an appendix. This correspondence suggests the possibility of understanding and generalising our results to a more abstract categorical setting.

The rest of the paper is structured as follows. In Section 2 we introduce the class of event structures we work with, i.e., poset event structures, and our reference behavioural equivalence, namely hereditary history-preserving bisimilarity. We also discuss how various event structure models in the literature embed into the considered class. In Section 3 we introduce and study the notion of folding, we prove the existence of a minimal quotient and we show the tight relation between general foldings and those on prime event structures. In Section 4 we present folding criteria on prime and asymmetric event structures, and discuss the existence of minimal quotients. Finally, in Section 5 we draw some conclusions, discuss connections with related literature and outline future work venues. An appendix discusses in detail the possibility of viewing foldings as open maps, the relation with abstraction homomorphisms and provides some results of technical nature.

This is an extended version of the conference paper [30]. Here we provide full proofs of the results, we slightly simplify the characterisation of foldings for PESs, we give a characterisation of foldings for asymmetric event structures and we treat in detail the relation with abstraction homomorphisms and the view of foldings as open maps.

2. Event Structures and History-Preserving Bisimilarity

In this section we define *hereditary history-preserving bisimilarity*, the reference behavioural equivalence in the paper. This is done for an abstract notion of event structure, introduced in [28], of which various stable event structure models in the literature can be seen as special subclasses. We will explicitly discuss prime [3], asymmetric [6], flow [7, 8] and bundle [9] event structures.

Notation. We first fix some basic notation on sets, relations and functions. Let $R \subseteq X \times X$ be a binary relation. Given $Y, Z \subseteq X$, we write $Y \ R^{\forall} \ Z$ (resp.

$Y\ R^{\exists}\ Z$) if for all (resp. for some) $y \in Y$ and $z \in Z$ it holds that $y\ R\ z$. When Y or Z are singletons, sometimes we replace them by their only element, writing, e.g., $y\ R^{\exists}\ Z$ for $\{y\}\ R^{\exists}\ Z$. The relation R is *acyclic* on Y if there is no $\{y_0, y_1, \ldots, y_n\} \subseteq Y$ such that $y_0\ R\ y_1\ R \ldots R\ y_n\ R\ y_0$. Relation R is a *partial order* if it is reflexive, antisymmetric and transitive. Given a function $f : X \to Y$ we will denote by $f[x \mapsto y] : X \cup \{x\} \to Y \cup \{y\}$ the function defined by $f[x \mapsto y](x) = y$ and $f[x \mapsto y](z) = f(z)$ for $z \in X \setminus \{x\}$. Note that this notation represents an update of f, when $x \in X$, or an extension of its domain, otherwise. For $Z \subseteq X$, we denote by $f_{|Z} : Z \to Y$ the restriction of f to Z.

2.1. Poset Event Structures

Following [28, 31, 32, 27], we work on a class of event structures where configurations are given as a primitive notion. More precisely, we borrow the idea of family of posets from [28].

Definition 2.1 (family of posets)**.** A *poset* is a pair $(C, \leq_C)$ where C is a set and $\leq_C$ is a partial order on C. A poset will be often denoted simply as C, leaving the partial order relation $\leq_C$ implicit. Given two posets C_1 and C_2 we say that C_1 is a *prefix* of C_2 and write $C_1 \sqsubseteq C_2$ if $C_1 \subseteq C_2$ and $\leq_{C_1} = \leq_{C_2} \cap (C_2 \times C_1)$. A *family of posets* F is a prefix-closed set of finite posets i.e., a set of finite posets such that if $C_2 \in F$ and $C_1 \sqsubseteq C_2$ then $C_1 \in F$. We say that two posets $C_1, C_2 \in F$ are compatible, written $C_1 \frown C_2$, if they have an upper bound, i.e., there is $C \in F$ such that $C_1, C_2 \sqsubseteq C$. The family of posets F is called *coherent* if each subset of F whose elements are pairwise compatible has an upper bound.

Posets C will be used to represent configurations, i.e., sets of events executed in a computation of an event structure. The order $\leq_C$ intuitively represents the order in which the events in C can occur. This motivates the notion of prefix order that can be interpreted as a computational extension: in order to have $C_1 \sqsubseteq C_2$ we require not only that $C_1 \subseteq C_2$, but also that

1. events in C_1 are ordered exactly as in C_2, i.e., the order in C_1 is the restriction of the order in C_2;
2. the new events in $C_2 \setminus C_1$ cannot precede events already in C_1 (i.e., for all $x_1 \in C_1$, $x_2 \in C_2$, if $x_2 \leq_{C_2} x_1$ then $x_2 \in C_1$).

While $\leq_{C_1} = \leq_{C_2} \cap (C_1 \times C_1)$ would be the right formalisation of (1) alone, requiring the stronger $\leq_{C_1} = \leq_{C_2} \cap (C_2 \times C_1)$ captures also (2).

An example of family of posets can be found in Fig. 1 (left). Observe, for instance, that the configuration with set of events $\{c\}$ is not a prefix of the one with set of events $\{a, c\}$, since in the latter $a \leq c$.

An event structure is then defined simply as a coherent family of posets where events carry a label. Hereafter Λ denotes a fixed set of labels.

Definition 2.2 (event structure)**.** A *(poset) event structure* is a tuple $\mathsf{E} = \langle E, \mathit{Conf}(\mathsf{E}), \lambda \rangle$ where E is a set of events, $\mathit{Conf}(\mathsf{E})$ is a coherent family of posets such that $E = \bigcup \mathit{Conf}(\mathsf{E})$ and $\lambda : E \to \Lambda$ is a labelling function. For a configuration $C \in \mathit{Conf}(\mathsf{E})$ the order $\leq_C$ is referred to as the *local order*.

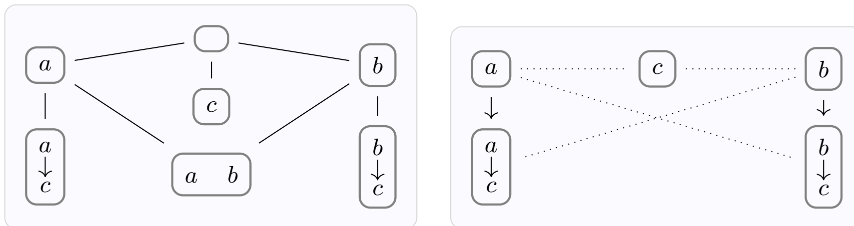


Figure 1: An event structure E and the canonical PES $\mathbb{P}(\mathsf{E})$

In [27] abstract event structures are defined as a collection of ordered configurations, without any further constraint. This is sufficient for giving some general definitions which are then studied in specific subclasses of event structures. Here, in order to develop a theory of foldings at the level of general event structures, we need to assume stronger properties, i.e, those of a family of posets from [28] (e.g, the fact that Definition 3.5 is well-given relies on this). This motivates the name poset event structure. Also note that, differently from what happens in other general concurrency models, like configuration structures [32], configurations are endowed explicitly with a partial order, which in turn intervenes in the definition of the prefix order between configurations. This will be essential to view event structures featuring asymmetric conflicts, like asymmetric event structures, as subclasses (see also Section 2.3).

Since we only deal with poset event structures and their subclasses, we will often omit the qualification "poset" and refer to them just as event structures. Moreover, we will often identify an event structure E with the underlying set E of events and write, e.g., $x \in \mathsf{E}$ for $x \in E$.

An *isomorphism of configurations* $f : C \to C'$ is an isomorphism of posets that respects the labels, i.e., for all $x, y \in C$, we have $\lambda(x) = \lambda(f(x))$ and $x \leq_C y$ iff $f(x) \leq_{C'} f(y)$. When configurations C, C' are isomorphic we write $C \simeq C'$.

As mentioned above, the prefix order on configurations can be interpreted as computational extension. This will be later formalised by a notion of transition system over the set of configurations (see Definition 2.4).

Given an event x in a configuration C it will be useful to refer to the prefix of C including only those events that necessarily precede x in C (and x itself). This motivates the following definition.

Definition 2.3 (history)**.** Let E be an event structure, let $C \in \mathit{Conf}(\mathsf{E})$ and let $x \in C$. The history of x in C is defined as the set $C[x] = \{y \in C \mid y \leq_C x\}$ endowed with the restriction of $\leq_C$ to $C[x]$, i.e., $\leq_{C[x]} = \leq_C \cap (C[x] \times C[x])$. The set of histories in E is $\mathit{Hist}(\mathsf{E}) = \{C[x] \mid C \in \mathit{Conf}(\mathsf{E}) \ \wedge \ x \in C\}$. The set of histories of a specific event $x \in \mathsf{E}$ will be denoted by $\mathit{Hist}(x)$.

Some properties of histories will be useful in the sequel.

Lemma 2.1 (properties of histories)**.** *Let* E *be an event structure. Then*

1. *for all* $C \in \mathit{Conf}(\mathsf{E})$*, we have* $C[x] \sqsubseteq C$*, hence* $C[x] \in \mathit{Conf}(\mathsf{E})$*;*

2. *for all* $C_1, C_2 \in \mathit{Conf}(\mathsf{E})$*,* $C_1 \sqsubseteq C_2$ *iff for all* $x \in C_1$*,* $C_1[x] = C_2[x]$*;*
3. *for all* $H_1, H_2 \in \mathit{Hist}(x)$*, if* $H_1 \frown H_2$ *then* $H_1 = H_2$*;*

Proof. 1. Immediate by the definition of $C[x]$.

2. Let $C_1, C_2 \in \mathit{Conf}(\mathsf{E})$ such that $C_1 \sqsubseteq C_2$. For all $x \in C_1$ we have that

$$\begin{aligned} C_2[x] &= \{y \in C_2 \mid y \leq_{C_2} x\} \\ &= \{y \in C_1 \mid y \leq_{C_1} x\} && [\text{since } C_1 \sqsubseteq C_2] \\ &= C_1[x] \end{aligned}$$

   Conversely, assume that for all $x \in C_1$ we have that $C_1[x] = C_2[x]$. Then, since $x \in C_i[x]$, for $i \in \{1, 2\}$, clearly $C_1 \subseteq C_2$. Moreover, for all $y \in C_1$ and $x \in C_2$, if $x \leq_{C_2} y$ then $x \in C_2[y]$. Therefore, since by hypothesis $C_1[y] = C_2[y]$, we have $x \in C_1$ and $x \leq_{C_1} y$, as desired. Therefore, $C_1 \sqsubseteq C_2$.
3. Let $H_1, H_2 \in \mathit{Hist}(x)$ and assume that $H_1 \frown H_2$. This means that there exists $C \in \mathit{Conf}(\mathsf{E})$ such that $H_1, H_2 \sqsubseteq C$. Therefore, by point (2), we have $H_1 = H_1[x] = C[x] = H_2[x] = H_2$.

□

In words, property (1) says that a history can be always extended to the full configuration it derives from. Property (2) means that a history of an event cannot change when the computation evolves. Finally, (3) states that different histories of the same event are incompatible.

2.2. Hereditary History-Preserving Bisimilarity

Hereditary history-preserving bisimilarity [23] is a classical equivalence in the true concurrency spectrum. In order to define it over poset event structures, it is convenient to have an explicit representation of the transitions between configurations.

Definition 2.4 (transition system)**.** Let E be an event structure. If $C, C' \in \mathit{Conf}(\mathsf{E})$ with $C \sqsubseteq C'$ we write $C \xrightarrow{X} C'$ where $X = C' \setminus C$.

When X is a singleton, i.e., $X = \{x\}$, we will often write $C \xrightarrow{x} C'$ instead of $C \xrightarrow{\{x\}} C'$. It is easy to see that in an event structure each configuration is reachable in the transition system from the empty one.

Lemma 2.2 (configurations are reachable)**.** *Let* E *be an event structure and let* $C \in \mathit{Conf}(\mathsf{E})$ *be a configuration. Then* $\emptyset \to^* C$*. More in detail, if* $x_1, x_2, \ldots, x_n$ *is any linearisation of* C *compatible with* $\leq_C$ *then, for all* $k \in \{1, \ldots, n\}$*,* $\{x_1, \ldots, x_{k-1}\} \xrightarrow{x_k} \{x_1, \ldots, x_{k-1}, x_k\}$ *.*

Proof. Immediate consequence of the prefix-closedness of the family of configurations. □

As it happens in the interleaving approach, a bisimulation between two event structures requires any event of an event structure to be simulated by an event of the other, with the same label. Additionally, the two events are required to have the same "causal history".

Definition 2.5 ((hereditary) history-preserving bisimilarity)**.** Let E, E' be event structures. A *history-preserving (hp-)bisimulation* is a set R of triples (C, f, C'), where $C \in \mathit{Conf}(\mathsf{E})$, $C' \in \mathit{Conf}(\mathsf{E}')$ and $f : C \to C'$ is an isomorphism of configurations, such that $(\emptyset, \emptyset, \emptyset) \in R$ and for all $(C_1, f, C_1') \in R$

1. for all $C_1 \xrightarrow{x} C_2$ there exists $C_1' \xrightarrow{x'} C_2'$ such that $(C_2, f[x \mapsto x'], C_2') \in R$;
2. for all $C_1' \xrightarrow{x'} C_2'$ there exists $C_1 \xrightarrow{x} C_2$ such that $(C_2, f[x \mapsto x'], C_2') \in R$.

Relation R is called a *hereditary history-preserving (hhp-)bisimulation* if, in addition, it is downward-closed, i.e., if $(C_1, f, C_1') \in R$ and $C_2 \subseteq C_1$ then $(C_2, f_{|C_2}, f(C_2)) \in R$.

Observe that, in the definition above, an event must be simulated by an event with the same label. In fact, in the triple $(C \cup \{x\}, f[x \mapsto x'], C' \cup \{x'\}) \in R$, the second component $f[x \mapsto x']$ must be an isomorphism of configurations, i.e., of labelled posets, and thus it preserves labels. Hhp-bisimilarity has been shown to arise as a canonical behavioural equivalence on prime event structures, as an instance of a general notion defined in terms of the concept of open map, when considering partially ordered computations as observations [25].

2.3. Examples: Prime, Asymmetric, Flow and Bundle Event Structures

We next observe how different kinds of event structures, introduced for various purposes in the literature, can be naturally viewed as subclasses of the poset event structures in Definition 2.2. This section is mainly intended to provide material for examples and discussions. The reader can quickly browse through it: only the correspondence with prime event structures will play a major role in the rest of the paper.

Prime event structures. Prime event structures [3] are one of the simplest and most popular event structure models, where dependencies between events are captured in terms of causality and conflict.

Definition 2.6 (prime event structure)**.** A *prime event structure* (PES, for short) is a tuple $\mathsf{P} = \langle E, \leq, \#, \lambda \rangle$, where E is a set of events, $\leq$ and $\#$ are binary relations on E called *causality* and *conflict*, respectively, and $\lambda : E \to \Lambda$ is a labelling function, such that

- $\leq$ is a partial order and $\lfloor x \rfloor = \{y \in E \mid y \leq x\}$ is finite for all $x \in E$;
- $\#$ is irreflexive, symmetric and hereditary with respect to causality, i.e., for all $x, y, z \in E$, if $x \# y$ and $y \leq z$ then $x \# z$.

The absence of conflicts between events is normally referred to as consistency. For later use, it is convenient to introduce a notation for it.

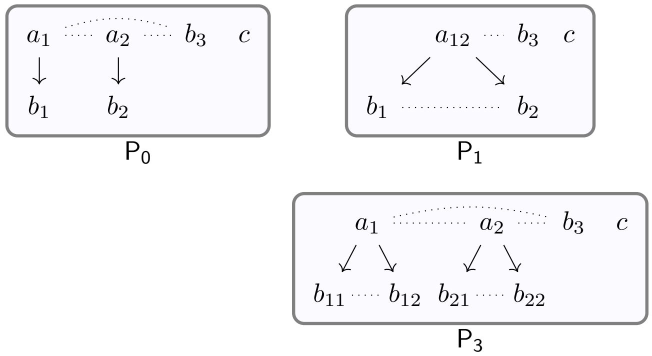


Figure 2: Some prime event structures

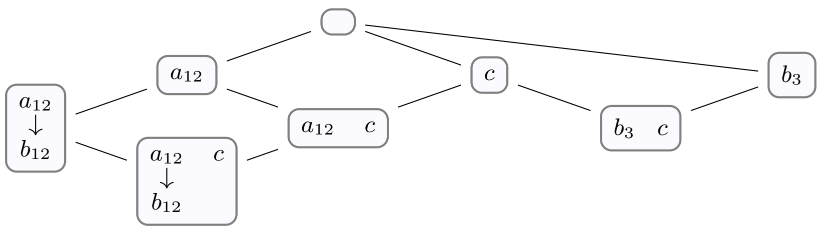


Figure 3: The configurations $Conf(\mathsf{P}_2)$ of the PES P_2 in Fig. 2 viewed as a poset event structure

Definition 2.7 (consistency)**.** Let $\mathsf{P} = \langle E, \leq, \#, \lambda \rangle$ be a PES. We say that $x, y \in E$ are *consistent*, written $x \frown y$, when $\neg(x \# y)$. A subset $X \subseteq E$ is called *consistent*, written $\frown X$, when its elements are pairwise consistent.

Configurations are consistent sets of events closed with respect to causality.

Definition 2.8 (PES configuration)**.** Let $\mathsf{P} = \langle E, \leq, \#, \lambda \rangle$ be a PES. A *configuration* of P is a finite set of events $C \subseteq E$ such that (i) for all $x \in C$, $\lfloor x \rfloor \subseteq C$ and (ii) $\frown C$.

Some examples of PESs can be found in Fig. 2. Causality is represented as a solid arrow, while conflict is represented as a dotted line. For instance, in P_0, event a_1 is a cause of b_1 and it is in conflict with both a_2 and b_3. Only direct causalities and non-inherited conflicts are represented. For instance, in P_0, the conflicts $a_1 \# b_2$, $a_2 \# b_1$, $b_1 \# b_2$, $b_1 \# b_3$ and $b_2 \# b_3$ are not represented since they are inherited. The labelling is implicitly represented by naming the events by their label, possibly with some index. E.g., a_1 and a_2 are events labelled by a.

Clearly PESs can be seen as poset event structures. Given a PES $\mathsf{P} = \langle E, \leq, \#, \lambda \rangle$ and its set of configurations $Conf(\mathsf{P})$, the local order of a configuration $C \in Conf(\mathsf{P})$ is $\leq_C = \leq \cap (C \times C)$, i.e., the restriction of the causality relation to C. The extension order turns out to be simply subset inclusion. In fact, given $C_1 \subseteq C_2$ clearly $\leq_{C_1} = \leq \cap (C_1 \times C_1)$ is the restriction to C_1 of $\leq_{C_2} = \leq$

$\cap(C_2 \times C_2)$. Moreover, if $x_1 \in C_1$ and $x_2 \in C_2$, with $x_2 \leq_{C_2} x_1$, then necessarily $x_2 \in C_1$ since configurations are causally closed. As an example, the PES P_2 of Fig. 2, viewed as a poset event structure, can be found in Fig. 3.

Asymmetric event structures. Asymmetric event structures [6] are a generalisation of PESs where conflict is allowed to be non-symmetric.

Definition 2.9 (asymmetric event structure)**.** An *asymmetric event structure* (AES, for short) is a tuple $\mathsf{A} = \langle E, \leq, \nearrow, \lambda \rangle$, where E is a set of events, $\leq$ and $\nearrow$ are binary relations on E called *causality* and *asymmetric conflict*, and $\lambda : E \to \Lambda$ is a labelling function, such that

- $\leq$ is a partial order and $\lfloor x \rfloor = \{y \in E \mid y \leq x\}$ is finite for all $x \in E$;
- $\nearrow$ satisfies, for all $x, y, z \in E$
  1. if $x < y$ then $x \nearrow y$;
  2. if $x \nearrow y$ and $y < z$ then $x \nearrow z$;
  3. $\nearrow$ is acyclic on $\lfloor x \rfloor$;
  4. if $\nearrow$ is cyclic on $\lfloor x \rfloor \cup \lfloor y \rfloor$ then $x \nearrow y$.

In the graphical representation, asymmetric conflict is depicted as a dotted arrow. For instance, in the asymmetric event structure A_0 of Fig. 4 we have $a_{12} \nearrow b_{123}$. Again, only non inherited asymmetric conflicts are represented.

The asymmetric conflict relation has two natural interpretations, i.e., $x \nearrow y$ can be understood as (i) the occurrence of y *prevents* x, or (ii) x *precedes* y in all computations where both appear. This allows us to represent faithfully the existence of precedences between actions and concurrent read accesses to a shared resource (intuitively, while readings can occur concurrently, destructive accesses can follow, but obviously not precede a reading).

The interpretation of asymmetric conflict above should give some intuition for the conditions in Definition 2.9. Condition (1) naturally arises from interpretation (ii) above: when $x < y$ clearly x precedes y when both occur and thus $x \nearrow y$. Condition (2) is a form of hereditarity of asymmetric conflict along causality: if $x \nearrow y$ and $y < z$ then all runs where x and z appear, necessarily also include y, and x precedes y which in turn precedes z, hence $x \nearrow z$. Concerning (3) and (4), observe that events forming a cycle of asymmetric conflict cannot appear in the same run, since each event in the cycle should occur before itself in the run. For instance, in the AES A_1 of Fig. 4, we have $a_1 \nearrow a_2 \nearrow a_1$, hence a_1 and a_2 cannot appear in the same computation. In this view, condition (3) corresponds to irreflexiveness of conflict in PESs, while condition (4) requires that binary symmetric conflict is explicitly represented by asymmetric conflict in both directions. Indeed, prime event structures can be identified with the subclass of AESs where $\nearrow$ is symmetric.

Configurations are again defined as sets of events which are causally closed and conflict free.

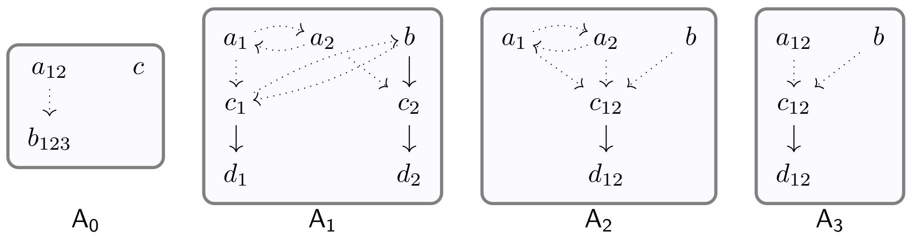


Figure 4: Some asymmetric event structures

Definition 2.10 (AES configuration)**.** Let $\mathsf{A} = \langle E, \leq, \nearrow, \lambda \rangle$ be an AES. A configuration of A is a finite set of events $C \subseteq E$ such that (i) for any $x \in C$, $\lfloor x \rfloor \subseteq C$ (causally closed) (ii) $\nearrow$ is acyclic on C (conflict free).

Also AESs can be seen as special poset event structures. Given an AES $\mathsf{A} = \langle E, \leq, \nearrow, \lambda \rangle$ and its set of configurations $\mathit{Conf}(\mathsf{A})$, the local order of a configuration $C \in \mathit{Conf}(\mathsf{A})$ is $\leq_C = (\nearrow \cap (C \times C))^*$, i.e., the transitive closure of the restriction of the asymmetric conflict to C. The prefix order on configurations is not simply set-inclusion: since a configuration C cannot be extended with an event which is prevented by some of the events already present in C. Hence for $C_1, C_2 \in \mathit{Conf}(\mathsf{A})$ we have $C_1 \sqsubseteq C_2$ iff $C_1 \subseteq C_2$ and for all $x \in C_1,\ y \in C_2 \setminus C_1$, $\neg(y \nearrow x)$. For instance, the configurations $\mathit{Conf}(\mathsf{A}_0)$ of the AES A_0 in Fig. 4, ordered by prefix, can be obtained from those of Fig. 3, by replacing all occurrences of b_{12} and b_3, by b_{123}. Note, e.g., that $\{b_{123}\} \not\sqsubseteq \{a_{12}, b_{123}\}$ since $a_{12} \nearrow b_{123}$.

Flow event structures. In some situations, it can be quite useful to have the possibility of modelling in a direct way the presence of multiple disjunctive and mutually exclusive causes for an event, something that is not possible in PESs and in AESs, where for each event there is a uniquely determined minimal set of causes. For instance, in a process calculus with non deterministic choice "+" and sequential composition ";" in order to give a PES semantics to $(a+b);c$ we are forced to use two different events to represent the execution of c, one for the execution of c after a and the other for the execution of c after b.

We briefly describe a model that overcomes this limitation, namely flow [7, 8] event structures.

Definition 2.11 (flow event structure)**.** A *flow event structure* (FES) is a tuple $\mathsf{F} = \langle E, \prec, \#, \lambda \rangle$, where E is a set of events, $\prec \subseteq E \times E$ is an irreflexive relation called the *flow relation*, $\# \subseteq E \times E$ is the *symmetric conflict* relation, and $\lambda : E \to \Lambda$ is a labelling function.

Causality is replaced by an irreflexive (in general non transitive) flow relation $\prec$, intuitively representing immediate causal dependency. Moreover, conflict is no longer hereditary.

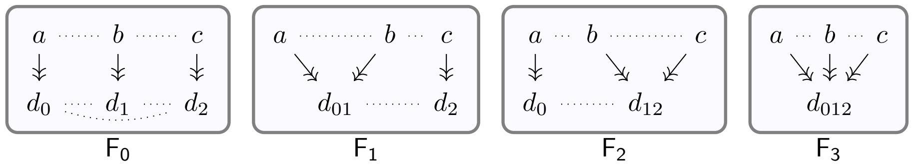


Figure 5: Some flow event structures

An event can have causes which are in conflict and these have a disjunctive interpretation, i.e., the event will be enabled by a maximal conflict-free subset of its causes. This is formalised by the notion of configuration.

Definition 2.12 (FES configuration)**.** Let $\mathsf{F} = \langle E, \prec, \#, \lambda \rangle$ be a FES. A configuration of F is a finite set of events $C \subseteq E$ such that (i) $\prec$ is acyclic on C, (ii) $\neg(x \# x')$ for all $x, x' \in C$ and (iii) for all $x \in C$ and $y \notin C$ with $y \prec x$, there exists $z \in C$ such that $y \# z$ and $z \prec x$.

Some examples of FESs can be found in Fig. 5. Relation $\prec$ is represented by a double headed solid arrow. For instance, consider the FES F_1. The set $C = \{a, d_{01}\}$ is a configuration. We have $b \prec d_{01}$ and $b \notin C$, but this is fine since there is $a \in C$ such that $a \# b$ and $a \prec d_{01}$.

Under mild assumptions that exclude the presence of non-executable events (a condition referred to as fullness in [8]), FESs can be seen as poset event structures, by endowing each configuration C with a local order arising as the reflexive and transitive closure of the restriction of the flow relation to C, i.e., $\leq_C = (\prec \cap (C \times C))^*$. Note that excluding the presence of non-executable events is necessary since in a poset event structure the set of events must coincide with the union of all configurations (condition $E = \bigcup \mathit{Conf}(\mathsf{E})$ in Definition 2.2) and all configurations are reachable (Lemma 2.2), hence all events are executable.

Bundle event structures. Bundle event structures [9, 14] are another event structure model that has been introduced in order to enable a direct representation of disjunctive causes, thus easing the definition of the semantics of the process description language LOTOS.

Definition 2.13 (bundle event structure)**.** A *bundle event structure*(BES) is a triple $\mathsf{B} = \langle E, \mapsto, \# \rangle$, where E is the set of events, $\# \subseteq E \times E$ is the irreflexive *symmetric conflict* relation and $\mapsto \subseteq \mathbf{2}^E_{\mathit{fin}} \times E$ is the *bundle relation* such that if $X \mapsto x$ then $X \times X \subseteq \#$.

Whenever $X \mapsto x$ the set X is called a *bundle* for the event x. It can be seen as a set of disjunctive and mutually exclusive causes for the event. The explicit representation of the bundles makes bundle event structures strictly less expressive than flow event structures, as briefly discussed below. On the other hand, bundle event structures offer the advantage of having a simpler theory. For instance, differently from what happens for flow event structures, non-executable events can be removed without affecting the behaviour of the event structure.

Configurations are conflict free sets of events such that if some event e is in the set then an event from of each of the bundles of e is also included in the set.

Definition 2.14 (BES configuration)**.** Let $\mathsf{B} = \langle E, \mapsto, \# \rangle$ be a BES. Let $\mapsto\!\!\!\!\twoheadrightarrow$ denote the binary relation on E defined, for $x, y \in E$, by $x \mapsto\!\!\!\!\twoheadrightarrow y$ if $X \mapsto y$ and $x \in X$. A finite set $C \subseteq E$ is a configuration if (i) the relation $\mapsto\!\!\!\!\twoheadrightarrow$ is acyclic on C (ii) $\neg(x \# y)$ for all $x, y \in C$; (iii) $X \cap C \neq \emptyset$ for all $x \in C$ and $X \mapsto x$.

Again, under the assumption that there are no non-executable events, one can turn a BES into a poset event structure in the sense of Definition 2.2 by endowing each configuration C with an order $\leq_C = (\mapsto\!\!\!\!\twoheadrightarrow \cap (C \times C))^*$.

Comparing models. It can be easily seen that the expressive power of AESs is incomparable with that of FESs and BESs. This is due to the fact that AESs cannot represent disjunctive causes, while FESs and BESs cannot represent asymmetric conflicts. For instance, consider the FES $\mathsf{F_4}$ in Fig. 6 (which can be also seen as a BES with a bundle $\{a, b\}$ for c). We have $\mathit{Conf}(\mathsf{F_4}) = \{\emptyset, \{a\}, \{b\}, \{a, c\}, \{b, c\}\}$ and it is easily seen that no AES exists with the same configurations. Now, consider the AES A in Fig. 6. We have $\mathit{Conf}(\mathsf{A}) = \{\emptyset, \{a\}, \{b\}, \{a, b\}\}$ and also here it is immediate to see that there is no FES or BES having the same configurations. In passing, we note also that this behavior cannot be expressed in terms of a configuration structure [32]. In fact, as shown in Fig. 6, in $\mathit{Conf}(\mathsf{A})$, the configuration $\{a, b\}$ is not an extension of $\{b\}$, consistently with the fact that a cannot be executed after b. Instead, in a configuration structure, where the order between configurations is just subset inclusion, configuration $\{a, b\}$ would be unavoidably seen as an extension of $\{b\}$.

In addition, as mentioned above and discussed in detail in [33, 14], BESs are strictly less expressive then FESs. For instance, for the FES $\mathsf{F_5}$ in Figure 6 there is no BES having the same configurations. This can be seen by observing that here only pairs of conflicting events can be in the same bundle.

Event structures models joining the features of AESs with those of FESs and BESs have been considered in the literature, like *extended* BESs [14], wich enrich BESs with asymmetric conflict, and FESs *with possible flow* [34, 35], which extend FESs with a possible flow relation enabling the representation of asymmetric conflicts. Also these generalised models can be viewed as poset event structures.

3. Foldings of Event Structures

In this section, we study a notion of folding, which is intended to formalise the intuition of a behaviour-preserving quotient for an event structure. We prove that there always exists a minimal quotient and we show that foldings between general poset event structures always arise, in a suitable formal sense, from foldings over prime event structures.

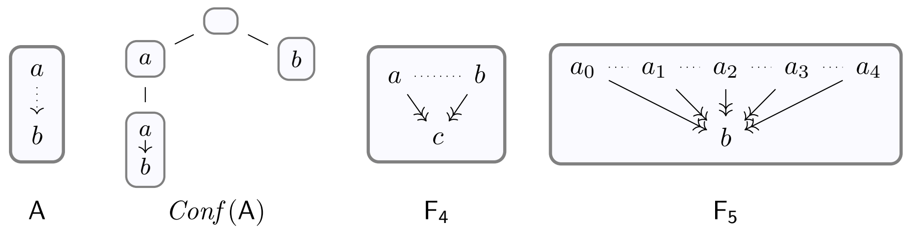


Figure 6: Separating examples for AESs, FESs, BESs

3.1. Morphisms and Foldings

We first endow event structures with a notion of morphism. Below, given two event structures E, E', a function $f : E \to E'$ and a configuration $C \in \mathit{Conf}(\mathsf{E})$, we write $f(C)$ to refer to the configuration whose underlying set is $\{f(x) \mid x \in C\}$, endowed with the order $f(x) \leq_{f(C)} f(y)$ iff $x \leq y$.

Definition 3.1 (morphism)**.** Let E, E' be event structures. A *(strong) morphism* $f : \mathsf{E} \to \mathsf{E}'$ is a function between the underlying sets of events such that $\lambda = \lambda' \circ f$ and for all configurations $C \in \mathit{Conf}(\mathsf{E})$, the function f is injective on C and $f(C) \in \mathit{Conf}(\mathsf{E}')$.

Hereafter, the qualification "strong" will be omitted since this is the only kind of morphisms we deal with. It is motivated by the fact that normally morphisms on event structures are designed to represent simulations. If this were the purpose, then the requirement on preservation of configurations could have been weaker, i.e., we could have asked the order in the target configuration to be included in (not identical to) the image of the order of the source configuration (precisely, given a configuration $\langle C, \leq_C \rangle \in \mathit{Conf}(\mathsf{E})$ then there exists $\langle C', \leq_{C'} \rangle \in \mathit{Conf}(\mathsf{E}')$ such that $C' = f(C)$ and for all $x, y \in C$, $f(x) \leq_{C'} f(y)$ implies $x \leq_C y$). Moreover, morphisms could have been partial. However, in our setting, for the objective of defining history-preserving quotients, the stronger notion works fine and simplifies the presentation.

Remark 1. *The composition of morphisms is a morphism and the identity is a morphism. Hence the class of event structures and event structure morphisms form a category* **ES***.*

Lemma 3.1 (morphisms preserve prefixes)**.** *Let E, E' be event structures, let $f : \mathsf{E} \to \mathsf{E}'$ be a morphism and let $C_1, C_2 \in \mathit{Conf}(\mathsf{E})$ be configurations. If $C_1 \sqsubseteq C_2$ then $f(C_1) \sqsubseteq f(C_2)$.*

Proof. Immediate, since from the definition of morphism we have $C_1 \simeq f(C_1)$ and $C_2 \simeq f(C_2)$. □

Definition 3.2 (folding)**.** Let E and E' be event structures. A *folding* is a morphism $f : \mathsf{E} \to \mathsf{E}'$ such that the relation $R_f = \{(C, f_{|C}, f(C)) \mid C \in \mathit{Conf}(\mathsf{E})\}$ is a hhp-bisimulation. A folding is called *elementary* if there is a set $X \subseteq \mathsf{E}$ such that for all $x, y \in \mathsf{E}$, $x \neq y$, we have $f(x) = f(y)$ if and only if $x, y \in X$.

In words, a folding is a function that "merges" some sets of events of an event structure into single events without altering the behaviour modulo hhp-bisimilarity. It is elementary if it merges only a single set of events. In [27] the notion of folding asks for the preservation of hp-bisimilarity, a weaker behavioural equivalence which is defined as hhp-bisimilarity but omitting the requirement of downward-closure. Note that, as far as the notion of folding is concerned, this makes no difference: R_f is downward-closed by definition, hence it is a hhp-bisimulation whenever it is a hp-bisimulation. Instead, taking hhp-bisimilarity as the reference equivalence appears to be the right choice for the development of the theory. E.g., it allows one to prove Lemma 3.8 that plays an important role for arguing about the adequateness of the notion of folding. Interestingly, foldings can be characterised as open maps in the sense of [25], by taking conflict free prime event structures as subcategory of observations. This is explicitly worked out in Appendix A.

As an example, consider the PESs in Fig. 2 and the function $f_{02} : \mathsf{P}_0 \to \mathsf{P}_2$ that maps events as suggested by the indices, i.e., $f_{02}(a_1) = f_{02}(a_2) = a_{12}$, $f_{02}(b_1) = f_{02}(b_2) = b_{12}$, $f_{02}(b_3) = b_3$ and $f_{02}(c) = c$. Then it is easy to see that f_{02} is a folding. Note that, instead, $f_{01} : \mathsf{P}_0 \to \mathsf{P}_1$, again mapping events according to their indices, is a morphism but not a folding. In fact, $f_{01}(\{a_1\}) = \{a_{12}\} \xrightarrow{b_2} \{a_{12}, b_2\}$, but clearly there is no transition $\{a_1\} \xrightarrow{x}$ with $f_{01}(x) = b_2$, since x can only be b_1 and the only counterimage of b_2 in P_0 is b_2.

It is also interesting to observe that the greater expressiveness of AESs allows one to obtain smaller quotients. For instance, while the PES P_2 in Fig. 2 is minimal in the class of PESs, if we view it as an AES, it can be further reduced. In fact the obvious function from P_2 to the AES A_0 in Fig. 4 can be easily seen to be a folding. Observe that this folding "transforms" the causality $a_{12} < b_{12}$ in P_2 into an asymmetric conflict $a_{12} \nearrow b_{123}$ in A_0. This is legal because also the event b_3 in P_2, in conflict with a_{12}, is mapped to b_{123}. In this way the the situation in which in A_0 the event b_{123} is executed before a_{12}, thus disabling this latter event, can be simulated in P_2 by executing b_3.

Lemma 3.2 (foldings are closed under composition)**.** *Let E, E', E'' be event structures and let $f : \mathsf{E} \to \mathsf{E}'$ and $f' : \mathsf{E}' \to \mathsf{E}''$ be foldings. Then $f' \circ f : \mathsf{E} \to \mathsf{E}''$ is a folding.*

Proof. We rely on the characterisation of foldings provided in Lemma 3.3. Let $C_1 \in \mathit{Conf}(\mathsf{E})$ and assume that $f'(f(C_1)) \xrightarrow{x''} C_2''$. Since $f(C_1) \in \mathit{Conf}(\mathsf{E}')$ and f' is a folding, there exists x' such that $f(C_1) \xrightarrow{x'} C_2'$ with $f'(x') = x''$ and $f'(C_2') = C_2''$. In turn, since f is a folding, from $f(C_1) \xrightarrow{x'} C_2'$, we derive the existence of a transition $C_1 \xrightarrow{x} C_2$ with $f(x) = x'$ and $f(C_2) = C_2'$. Therefore $f'(f(x)) = x''$ and $f'(f(C_2)) = C_2''$, as desired. □

Remark 2. *Since composition of foldings is a folding (Lemma 3.2) and the identity is a folding, we can consider a subcategory $\mathbf{ES_f}$ of $\mathbf{ES}$ with the same objects and foldings as morphisms.*

Again in the setting of AESs, consider the structures in Fig. 4 and the functions $g_{12} : \mathsf{A}_1 \to \mathsf{A}_2$, and $g_{23} : \mathsf{A}_2 \to \mathsf{A}_3$, naturally induced by the indices. These can be seen to be foldings. The first one merges c_1, in conflict with b and c_2 caused by b to a single event c_{12}, in asymmetric conflict with b. The second one merges the two conflicting events a_1 and a_2 into a single one a_{12}. Their composition $g_{13} = g_{23} \circ g_{12} : \mathsf{A}_1 \to \mathsf{A}_3$ is again a folding.

As a last example, consider the FESs in Fig. 5. Again the obvious functions from F_0 to F_1 and F_2 can be seen to be foldings. Instead, seen as a PES, the event structure F_0 is minimal.

The next result shows that if we know that $f : \mathsf{E} \to \mathsf{E}'$ is a morphism, then half of the conditions needed to be a hhp-bisimulation and thus a folding, i.e., condition (1) in Definition 2.5, is automatically satisfied. This is used later in proofs whenever we need to show that some map is a folding.

Lemma 3.3 (from morphisms to foldings). *Let E and E' be event structures and let $f : \mathsf{E} \to \mathsf{E}'$ be a morphism. If for all $C_1 \in \mathit{Conf}(\mathsf{E})$ and transition $f(C_1) \xrightarrow{x'} C_2'$ there exists $C_1 \xrightarrow{x} C_2$ such that $f(C_2) = C_2'$ then f is a folding.*

Proof. We have to show that $R_f = \{(C, f_{|C}, f(C)) \mid C \in \mathit{Conf}(\mathsf{E})\}$ satisfies conditions (1) and (2) of Definition 2.5. Condition (2) is in the hypotheses. Concerning (1), let $C_1 \in \mathit{Conf}(\mathsf{E})$ and consider a transition $C_1 \xrightarrow{x} C_2$. Then by definition of morphism, $f(C_i)$ is in $\mathit{Conf}(\mathsf{E})$ and it is isomorphic to C_i, for $i \in \{1, 2\}$. Therefore $f(C_1) \xrightarrow{f(x)} f(C_2)$. □

Another simple but crucial result shows that the target event structure for a folding is completely determined by the mapping on events. This allows us to view foldings as equivalences on the source event structures. We first define the quotient induced by a morphism.

Definition 3.3 (quotients from morphisms). Let E, E' be event structures and let $f : \mathsf{E} \to \mathsf{E}'$ be a morphism. Let $\equiv_f$ be the equivalence relation on E defined by $x \equiv_f y$ if $f(x) = f(y)$. We denote by $\mathsf{E}_{/\equiv_f}$ the event structure with configurations $\mathit{Conf}(\mathsf{E}_{/\equiv_f}) = \{[C]_{\equiv_f} \mid C \in \mathit{Conf}(\mathsf{E})\}$ where $[C]_{\equiv_f} = \{[x]_{\equiv_f} \mid x \in C\}$ is ordered by $[x]_{\equiv_f} \leq_{[C]_{\equiv_f}} [y]_{\equiv_f}$ iff $x \leq_C y$.

It is immediate to see that $\mathsf{E}_{/\equiv_f}$ is a well-defined event structure. This also follows from the lemma below.

Lemma 3.4 (folding as equivalences). *Let E, E' be event structures and let $f : \mathsf{E} \to \mathsf{E}'$ be a morphism. If f is a folding then $\mathsf{E}_{/\equiv_f}$ is isomorphic to E'.*

Proof. Consider the function $g : E_{/\equiv_f} \to \mathsf{E}'$ defined by $g([x]_{\equiv_f}) = f(x)$. It is well defined, since all elements in $[x]_{\equiv_f}$ have the same f-image, and clearly injective. Moreover, it is also surjective. In fact, if $x' \in \mathsf{E}'$ then there exists $C' \in \mathit{Conf}(\mathsf{E}')$ such that $x' \in C'$. By Lemma 2.2, configuration C' is reachable from the empty one, and thus, since f is an hp-bisimulation, there exists $C \in \mathit{Conf}(\mathsf{E})$ such that $C' = f(C)$. Therefore there is $x \in C$ such that $f(x) = x'$ and thus $g([x]_{\equiv_f}) = x'$.

Finally, observe that by definition, for all configuration $C' \in \mathit{Conf}(\mathsf{E}_{/\equiv_f})$, we have $g(C') \simeq C'$, hence we conclude. ☐

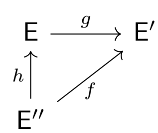
The previous result allows us to identify foldings with the corresponding equivalences on the source event structures and motivates the following definition.

Definition 3.4 (folding equivalences)**.** Let E be an event structure. The set of folding equivalences over E is $\mathit{FEq}(\mathsf{E}) = \{\equiv_f \mid f : \mathsf{E} \to \mathsf{E}' \text{ folding for some } \mathsf{E}'\}$.

Hereafter, we will freely switch between the two views of foldings as morphisms or as equivalences, since each will be convenient for some purposes.

We next observe that given two foldings we can always take their "join", providing a new folding that, roughly speaking, produces a smaller quotient than both the original ones. We first show a useful factorisation property involving morphisms and foldings.

Lemma 3.5 (factorising morphisms)**.** *Let E, E', E'' be event structures and let $f : \mathsf{E}'' \to \mathsf{E}'$ be a morphism and $h : \mathsf{E}'' \to \mathsf{E}$ be a folding. Let $g : \mathsf{E} \to \mathsf{E}'$ be a function such that $f = g \circ h$.*



Then g is a morphism. Moreover, if f is a folding then g is.

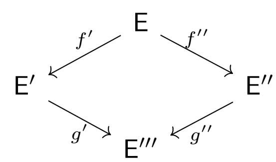
Proof. Let us show that g is a morphism. For all $C \in \mathit{Conf}(\mathsf{E})$, since h is a folding, there exists $C'' \in \mathit{Conf}(\mathsf{E}'')$ such that $h(C'') = C$ and $C'' \simeq C$. Since f is a morphism $f(C'') \in \mathit{Conf}(\mathsf{E}')$. Therefore $g(C) = g(h(C'')) = f(C'')$, as desired.

Let us assume now that g is a folding. Let $C_1 \in \mathit{Conf}(\mathsf{E})$ and suppose that there is a transition $g(C_1) \xrightarrow{x'} C_2'$. Since h is a folding, there is a configuration $C_1'' \in \mathit{Conf}(\mathsf{E}'')$ such that $C_1 = h(C_1'')$. Therefore $f(C_1'') = g(h(C_1'') = g(C_1) \xrightarrow{x'} C_2'$. Since f is a folding there is a transition $C_1'' \xrightarrow{x''} C_2''$ with $f(C_2'') = C_2'$. Therefore $h(C_2'') = C_1 \xrightarrow{h(x'')} h(C_2'')$ with $g(h(C_2'')) = f(C_2) = C_2'$, as desired. ☐

We can then prove the desired result concerning the possibility of joining foldings.

Proposition 3.6 (joining foldings)**.** *Let $\mathsf{E}, \mathsf{E}', \mathsf{E}''$ be event structures and let $f' : \mathsf{E} \to \mathsf{E}'$, $f'' : \mathsf{E} \to \mathsf{E}''$ be foldings. Define E''' as the quotient $\mathsf{E}_{/\equiv}$ where $\equiv$ is the transitive closure of $\equiv_{f'} \cup \equiv_{f''}$. Then $g' : \mathsf{E}' \to \mathsf{E}'''$ defined by $g'(x') = [x]_\equiv$ if $f'(x) = x'$ and $g'' : \mathsf{E}'' \to \mathsf{E}'''$ defined by $g''(x'') = [x]_\equiv$ if $f''(x) = x''$ are foldings.*

Proof. We actually show that the construction described in the statement produces the pushout in the category **ES** and also in $\mathbf{ES_f}$. Consider the diagram



Observe that E''', with functions g' and g'' is the pushout in **Set**, as it easily follows recalling that f' and f'' are surjective. Another immediate observation is that the set of configurations of E''' can be written

$$Conf(\mathsf{E}''') = \{g'(f'(C)) \mid C \in Conf(\mathsf{E})\} = \{g''(f''(C)) \mid C \in Conf(\mathsf{E})\} \quad (1)$$

We prove that g' is a folding. In fact

- g' is a morphism.
  For all $C' \in Conf(\mathsf{E}')$, since f' is a folding, there is $C \in Conf(\mathsf{E})$ such that $f'(C) = C'$. Therefore $g'(C') = g'(f'(C)) \in Conf(\mathsf{E}''')$, by construction. Moreover, g' is injective on C'. In fact, take $x', y' \in C'$, with $g'(x') = g'(y')$. Since $C' = f'(C)$, there are $x, y \in C$ such that $f'(x) = x'$ and $f'(y) = y'$. Therefore, $g'(f'(x)) = g'(f'(y))$, and thus, by the properties of pushouts, $f''(x) = f''(y)$. Since f'' is a folding, thus a morphism, this implies $x = y$ and thus $x' = f'(x) = f'(y) = y'$, as desired.

- g' is a folding.
  Let $C'_1 \in Conf(\mathsf{E}')$ and assume that $f'(C'_1) \xrightarrow{x'''} D'''_2$. By (1) we know that there is $D_2 \in Conf(\mathsf{E})$ such that $D'''_2 = g'(f'(D_2))$ and $D_2 \simeq D'''_2$. Therefore, there is $D_1 \sqsubseteq D_2$ such that $f'(g'(D_1)) = g'(C'_1)$ and

  $$D_1 \xrightarrow{x} D_2. \quad (2)$$

  Define $D'_1 = f'(D_1) \in Conf(\mathsf{E}')$. Now, since f' is a folding and $C'_1 \in Conf(\mathsf{E}_1)$, there is also $C_1 \in Conf(\mathsf{E})$ such that $f'(C_1) = C'_1$. Recall that $g'(D'_1) = f'(g'(D_1)) = g'(C'_1)$, hence, by pushout properties, it must be $f''(C_1) = f''(D_1)$. From (2), since f'' is a folding, we deduce $f''(C_1) = f''(D_1) \xrightarrow{x''} D''_2$, with $f''(D_2) = D''_2$. And, using again the fact that f'' is a folding, this implies $C_1 \xrightarrow{y} C_2$, with $f''(C_2) = D''_2 = f''(D_2)$.

  Now, we use the fact that f' is a folding, and derive that $C'_1 = f'(C_1) \xrightarrow{f'(x_1)} f'(C_2)$. If we call $C'_2 = f'(C_2)$, we have that $g'(C'_2) = g'(D'_2)$, as desired, since $f''(C_2) = f''(D_2)$.

In the same way, one concludes that also g'' is a folding.

Given any other E_1 with morphisms $g'_1 : \mathsf{E}' \to \mathsf{E}_1$ and $g''_1 : \mathsf{E}'' \to \mathsf{E}_1$ such that $g'_1 \circ f' = g'_2 \circ f''$, we show that there exists a unique morphism $h : \mathsf{E}''' \to \mathsf{E}_1$ that makes the diagram commute.

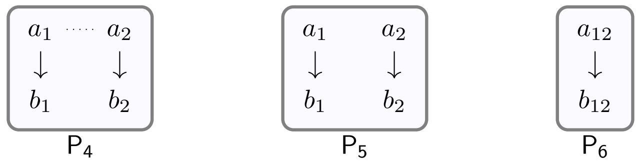
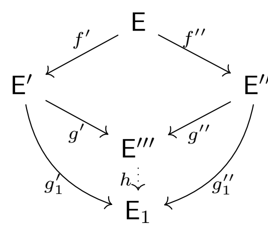


Figure 7: Non existence of pushout of general morphisms



Consider the unique map $h : \mathsf{E}''' \to \mathsf{E}_1$ making the diagram commute in **Set**. Since g' is a folding and g'_1 is a morphism, by Lemma 3.5, also h is a morphism. This proves that E''' is a pushout in **ES**.

By the same result, if g'_1 is a folding, also the mediating morphism h is. This means that the same construction produces a pushout in $\mathbf{ES_f}$. □

As an example, consider the PES in Fig. 2 and two morphisms $f_{30} : \mathsf{P}_3 \to \mathsf{P}_0$ and $f_{31} : \mathsf{P}_3 \to \mathsf{P}_1$. The way all events are mapped by f_{30} and f_{31} is naturally suggested by their labelling, apart from the events b_{ij} for which we let $f_{30}(b_{ij}) = b_i$ while $f_{31}(b_{ij}) = b_j$. It can be seen that both are foldings. Their join, constructed as in Proposition 3.6, is P_2 with the folding morphisms $f_{02} : \mathsf{P}_0 \to \mathsf{P}_2$ and $f_{12} : \mathsf{P}_1 \to \mathsf{P}_2$.

Remark 3. *Proposition 3.6 is a consequence of the fact that the category* **ES** *has pushouts of foldings. Indeed,* E''' *as defined in Proposition 3.6 is the pushout of* f' *and* f'' *(in* **ES** *and also in* $\mathbf{ES_f}$*).*

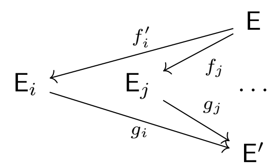
Also note that **ES** *does not have all pushouts. As a counterexample to the existence of pushouts in* **ES** *for general morphisms, consider the obvious mappings* $f_{45} : \mathsf{P}_4 \to \mathsf{P}_5$ *and* $f_{46} : \mathsf{P}_4 \to \mathsf{P}_6$ *in Fig. 7. It is easy to realise that, if a pushout existed, the mapping from* P_5 *into the pushout object should identify the concurrent events* a_1 *and* a_2*, failing to be an event structure morphism.*

When interpreted in the set of folding equivalences of an event structure, Proposition 3.6 has a clear meaning. Recall that the equivalences over some fixed set X, ordered by inclusion, form a complete lattice, where the top element is the universal equivalence $X \times X$ and the bottom is the identity on X. Then Proposition 3.6 implies that $FEq(\mathsf{E})$ is a sublattice of the lattice of equivalences. Actually, it can be shown that $FEq(\mathsf{E})$ is itself a complete lattice. This implies that each event structure E admits a maximally folded version.

Theorem 3.7 (lattice of foldings)**.** *Let* E *be an event structure. Then* $\mathit{FEq}(\mathsf{E})$ *is a sublattice of the complete lattice of equivalence relations over* E*.*

Proof. We proceed by showing a generalisation of Proposition 3.6 from which the thesis follows. We prove that for any event structure E, each collection of foldings $f_i : \mathsf{E} \to \mathsf{E}_i$, with $i \in I$, admits a colimit in $\mathbf{ES}$.

When I is finite, the proof proceeds by straightforward induction on I, using Proposition 3.6. If instead I is infinite, let E' be the colimit of the f_i's in $\mathbf{Set}$.



with configurations $\mathit{Conf}(\mathsf{E}') = \{g_i(f_i(C)) \mid C \in \mathit{Conf}(\mathsf{E})\}$. The proof of the fact that the g_i's are foldings then proceeds as in Proposition 3.6. The only delicate point is the following. Given configurations $C, C' \in \mathit{Conf}(\mathsf{E})$, define $C_i = f(C)$ and $C'_i = f(C') \in \mathit{Conf}(\mathsf{E}_i)$. If $g_i(C_i) = g_i(C'_i)$, then it is not necessarily the case that $f_j(C) = f_j(C')$ for some $j \in I$. However, since configurations are finite, there is a finite subset $J \subseteq I$ such that, if E_J is the colimit of $\{f_j \mid j \in J\}$ and $f_J : \mathsf{E} \to \mathsf{E}_J$ the corresponding folding, whose existence is proved in the first part, then $f_J(C) = f_J(C')$. Exploiting this fact, we can conclude exactly as in Proposition 3.6. $\square$

Remark 4. *The proof of Theorem 3.7 shows that for any event structure* E*, each collection of foldings* $f_i : \mathsf{E} \to \mathsf{E}_i$*, with* $i \in I$*, admits a colimit in* $\mathbf{ES}$*. Thus the coslice category* $(\mathsf{E} \downarrow \mathbf{ES_f})$ *has a terminal object, which is the maximally folded event structure.*

It is natural to ask whether all behaviour preserving quotients correspond to foldings. Strictly speaking, the answer is negative. More precisely, there can be morphisms $f : \mathsf{E} \to \mathsf{E}'$ such that $\mathsf{E}_{/\equiv_f}$ is hhp-bisimilar to E, but f is not a folding. For an example, consider the PESs P_0 and P_1 in Fig. 2 and the morphism $f_{01} : \mathsf{P}_0 \to \mathsf{P}_1$ suggested by the indexing. We already observed this is not a folding, but $\mathsf{P}_{0/\equiv_{f_{01}}}$, which is isomorphic to P_1, is hhp-bisimilar to P_0.

However, we can show that for any behaviour preserving quotient, there is a folding that produces a coarser equivalence, and thus a smaller quotient. For instance, in the example discussed above, there is the folding $f_{02} : \mathsf{P}_0 \to \mathsf{P}_2$, that "produces" a smaller quotient.

This follows from two results. The first one is the possibility of joining foldings (Proposition 3.6). The second one, proved below, is the possibility of viewing a hhp-bisimulation between two event structures E', E'' as an event structure itself. This is a generalisation to our setting of a property proved for PESs in [23].

Lemma 3.8 (hhp-bisimulation as an event structure)**.** *Let* E'*,* E'' *be event structures and let* R *be a hhp-bisimulation between them. Then there exists a (prime) event structure* E_R *and two foldings* $\pi' : \mathsf{E}_R \to \mathsf{E}'$ *and* $\pi'' : \mathsf{E}_R \to \mathsf{E}''$*.*

Proof. Let E', E'' be event structures and let R be a hhp-bisimulation between them. Define E_R as follows. Events are histories related by R, namely the triples $\{(H', f, H'')\} \mid H' \in \mathit{Hist}(\mathsf{E}')\}$, labelled by $\lambda_{\mathsf{E}_R}(H', f, H'') = \lambda_{\mathsf{E}}(x')$ when $H' \in \mathit{Hist}(x')$. For each $(C', f, C'') \in R$, define

$$C_f = \{(C'[x'], f_{|C'[x']}, C''[f(x')]) \mid x \in C'\}$$

ordered by pointwise inclusion, i.e., $(H'_1, f_1, H''_1) \leq_{C_f} (H'_2, f_2, H''_2)$ if $f_1 \subseteq f_2$, and thus $H'_1 \subseteq H''_1$, $H'_2 \subseteq H''_2$. The set of configurations of E_R is $\mathit{Conf}(\mathsf{E}_R) = \{C_R \mid C \in \mathit{Conf}(\mathsf{E})\}$.

It is easy to see that $\mathit{Conf}(\mathsf{E}_R)$ is well-defined. Prefix-closedness of $\mathit{Conf}(\mathsf{E}_R)$ follows from the fact that R is downward-closed by definition of hhp-bisimulation. It can be seen that E_R is actually a prime event structure, with causality defined by $(H'_1, f_1, H''_1) \leq (H'_2, f_2, H''_2)$ if $H'_1 \sqsubseteq H'_2$ and $f_1 \sqsubseteq f_2$, and conflict defined by $(H'_1, f_1, H''_1) \# (H'_2, f_2, H''_2)$ if there is no $(C', f, C'') \in R$ such that $H'_1, H'_2 \sqsubseteq C'$ and $f_1, f_2 \subseteq f$.

Consider two configurations $C_{f_1}, C_{f_2} \in \mathit{Conf}(\mathsf{E}_R)$, arising from the triples $(C'_i, f_i, C''_i) \in R$, for $i \in \{1, 2\}$. Then it holds that

$C_{f_1} \sqsubseteq C_{f_2}$
iff $C_{f_1} \subseteq C_{f_2}$
iff for all $x' \in C'_1$, $(C'_1[x'], f_{1|C'_1[x']}, C''_1[f_1(x')]) \in C_{f_2}$
iff for all $x' \in C'_1$, $C'_1[x'] = C'_2[x']$ and $f_1(x') = f_2(x')$
iff $C'_1 \sqsubseteq C'_2$ and $f_1 \subseteq f_2$.

We can now define $\pi' : \mathsf{E}_R \to \mathsf{E}'$ as $\pi'(H', f, H'') = x'$ if $H' \in \mathit{Hist}(x')$ and, similarly, $\pi'' : \mathsf{E}_R \to \mathsf{E}'$ as $\pi''(H', f, H'') = x''$ if $H'' \in \mathit{Hist}(x'')$.

Then π' and π'' are well-defined morphisms and they are foldings. We prove this for π' (for π'' the proof is completely analogous).

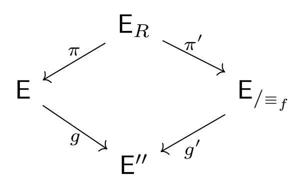
- π' is a morphism.
  This is immediate by observing that for any configuration $C_f \in \mathit{Conf}(\mathsf{E}_R)$, arising from the triple $(C', f, C'') \in R$, then we have $\pi'(C_f) = C'$. Note that, concerning the local order, for $x', y' \in C'$ we have $(C'[x'], f_{|C'[x']}, C''[f(x')]) \leq_{C_f} (C'[y'], f_{|C'[y']}, C''[f(y')])$ iff inclusion holds pointwise iff $x' \in C'[y']$ iff $x' \leq_{C'} y'$, which means $\pi'(C'[x']) = x' \leq_{C'} y' = \pi'(C'[y'])$.
- π' is a folding.
  In fact, for any configuration $C_f \in \mathit{Conf}(\mathsf{E}_R)$, arising from the triple $(C', f, C'') \in R$, if $\pi'(C_f) = C' \xrightarrow{x'} D'$ then, since R is an hhp-bisimulation, there is $C'' \xrightarrow{x''} D''$ with $(C'', g, D'') \in R$ with $g = f[x' \mapsto x'']$. Hence, if we let $H' = D'[x']$, we have that $C_f \xrightarrow{(H', g_{|H'}, g(H'))} C_g$ and $\pi'(C_g) = D'$, as desired.

□

We can finally prove the desired property.

Proposition 3.9 (foldings subsume behavioural quotients)**.** *Let E be an event structure and let $f : \mathsf{E} \to \mathsf{E}'$ be a morphism such that $\mathsf{E}_{/\equiv_f}$ is hhp-bisimilar to E. Then there exists a folding $g : \mathsf{E} \to \mathsf{E}''$ such that $\equiv_g$ is coarser than $\equiv_f$.*

Proof. Let R be a hhp-bisimulation between E and $\mathsf{E}_{/\equiv_f}$. Consider the event structure E_R and the foldings $\pi : \mathsf{E}_R \to \mathsf{E}$ and $\pi' : \mathsf{E}_R \to \mathsf{E}_{/\equiv_f}$, given by Lemma 3.8. By Proposition 3.6 we can close the diagram as follows:



and both g and g' are foldings. Then $\mathsf{E}'' = \mathsf{E}_{/\equiv_g} = \left(\mathsf{E}_{/\equiv_f}\right)_{/\equiv_{g'}}$ and we conclude.

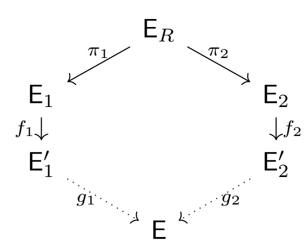
□

We already proved in Theorem 3.7 that every event structure admits a maximally folded version. Relying on Lemma 3.8 we can also prove that the maximally folded versions of hhp-bisimilar event structures are isomorphic, i.e., there is a unique minimal quotient for each hhp-equivalence class of event structures.

Corollary 3.10 (unique minimal quotient)**.** *Let E_1 and E_2 be hhp-bisimilar event structures and let E'_1 and E'_2 be the corresponding maximally folded versions. Then E'_1 and E'_2 are isomorphic.*

Proof. Let R be a hhp-bisimulation between E_1 and E_2. By Lemma 3.8 we can turn R into a (prime) event structure E_R with two foldings $\pi_1 : \mathsf{E}_R \to \mathsf{E}_1$ and $\pi_2 : \mathsf{E}_R \to \mathsf{E}_2$. Let $f_1 : \mathsf{E}_1 \to \mathsf{E}'_1$ and $f_2 : \mathsf{E}_2 \to \mathsf{E}'_2$ be the folding morphism of E_1 and E_2 into their maximally folded versions given by Theorem 3.7.

By Proposition 3.6 we can obtain the following pushout diagram



For $i \in \{1, 2\}$, since $f_i \circ \pi_i : \mathsf{E_i} \to \mathsf{E}$ is a folding and E'_i is final in $(\mathsf{E_i} \downarrow \mathbf{ES_f})$ we deduce that E'_i is isomorphic to E. Hence E'_1 and E'_2 are isomorphic, as desired. □

Remark 5. *Corollary 3.10, in categorical terms, shows that for every event struture* E*, the full subcategory of* $\mathbf{ES_f}$ *having as objects the event structures in the hhp-bisimilarity class of* E *has a terminal object, which is the (common) maximally folded event structure for the event structures in the class.*

3.2. Folding through Prime Event Structures

Here we observe that each poset event structure is the folding of a corresponding canonical PES. We then prove that, interestingly enough, all foldings between event structures arise from foldings of the corresponding canonical PESs.

We start with the definition of the canonical PES associated with an event structure.

Definition 3.5 (PES for an event structure)**.** Let E be an event structure. Its canonical PES is $\mathbb{P}(\mathsf{E}) = \langle \mathit{Hist}(\mathsf{E}), \sqsubseteq, \#, \lambda' \rangle$ where $\sqsubseteq$ is prefix, $\#$ is inconsistency, i.e., for $H_1, H_2 \in \mathit{Hist}(\mathsf{E})$ we let $H_1 \# H_2$ if $\neg(H_1 \frown H_2)$ and $\lambda'(H) = \lambda(x)$ when $H \in \mathit{Hist}(x)$. Given a morphism $f : \mathsf{E} \to \mathsf{E}'$ we write $\mathbb{P}(f) : \mathbb{P}(\mathsf{E}) \to \mathbb{P}(\mathsf{E}')$ for the morphism defined by $\mathbb{P}(f)(H) = f(H)$.

It can be easily seen that the definition above is well-given. In particular, $\mathbb{P}(\mathsf{E})$ is a well-defined PES because, as proved in [28], a family of posets ordered by prefix is a finitary coherent prime algebraic domain. Then the tight relation between this class of domains and PESs highlighted in [3] allows one to conclude the proof. For instance, in Fig. 1(right) one can find the canonical PES for the event structure on the left.

The canonical PES associated with an event structure can always be folded to the original event structure. For this purpose, it is useful to state some properties of the corresponding partial orders of configurations.

Lemma 3.11 (configurations of the canonical PES)**.** *Let* E *be an event structure. Then* $\mathit{Conf}(\mathsf{E})$ *and* $\mathit{Conf}(\mathbb{P}(ES))$ *seen as partial orders, ordered by prefix, are isomorphic.*

More in detail, for all $C \in \mathit{Conf}(\mathsf{E})$ *it holds* $hs(C) = \{C[x] \mid x \in C\}$*, with inclusion as local order, is in* $\mathit{Conf}(\mathbb{P}(\mathsf{E}))$*. Moreover* $C \simeq hs(C)$ *and* $hs(\cdot) : \mathit{Conf}(\mathsf{E}) \to \mathit{Conf}(\mathbb{P}(\mathsf{E}))$ *is a poset isomorphism.*

Its inverse is as follows. For $D \in \mathit{Conf}(\mathbb{P}(\mathsf{E}))$ *consider* $\mathit{fl}(D) = \bigcup D$*. Then, for each* $x \in \mathit{fl}(D)$ *there exists a unique* $H_x \in D$ *such that* $H_x \in \mathit{Hist}(x)$*. Define the order* $\leq_{\mathit{fl}(D)}$*, for* $x, y \in \mathit{fl}(D)$*, by* $x \leq_{\mathit{fl}(D)} y$ *iff* $x \in H_y$*. Then* $\mathit{fl}(D) \in \mathit{Conf}(\mathsf{E})$ *and* $\mathit{fl}(D) \simeq D$ *as posets.*

Proof. Let $C \in \mathit{Conf}(\mathsf{E})$ and let us show that $hs(C) = \{C[x] \mid x \in C\}$, with inclusion as local order, is in $\mathit{Conf}(\mathbb{P}(\mathsf{E}))$. First, note that $hs(C)$ is consistent by construction, since $C[x] \sqsubseteq C$ for all $x \in C$. Moreover, it is causally closed. In fact, if $H \sqsubseteq C[x]$ for some $H \in \mathit{Hist}(\mathsf{E})$, then, if $H \in \mathit{Hist}(y)$, by Lemma 2.1(2) we have $H = C[x][y] = C[y] \in hs(C)$. Moreover, $hs(C)$ is isomorphic to C, the isomorphism established by the mapping $C[x] \mapsto x$. It is clearly bijective. Moreover, for all $x_1, x_2 \in C$ it holds that $C[x_1] \subseteq C[x_2]$ iff $x_1 \in C[x_2]$ and thus $x_1 \leq_C x_2$.

Let us show that $hs(\cdot) : \mathit{Conf}(\mathsf{E}) \to \mathit{Conf}(\mathbb{P}(\mathsf{E}))$ is a poset isomorphism. It is injective. In fact, if $hs(C_1) = hs(C_2)$ then clearly C_1 and C_2 contain the same events. Moreover, $\leq_{C_1} = \leq_{C_2}$ and thus the two configurations coincide. Otherwise, there would be $x, y \in C_1$ such that $x \leq_{C_1} y$ and $\neg(x \leq_{C_2} y)$, or conversely $\neg(x \leq_{C_1} y)$ and $x \leq_{C_2} y$. Assume, without loss of generality, that we are in the first case. Then $x \in C_1[y]$ and $x \notin C_2[y]$, and thus $hs(C_1) \neq hs(C_2)$ contradicting the hypotheses. Moreover, it preserves and reflects the prefix order, i.e., given $C_1, C_2 \in \mathit{Conf}(\mathsf{E})$ we have $C_1 \sqsubseteq C_2$ iff $hs(C_1) \subseteq hs(C_2)$ as it immediately follows from Lemma 2.1(2).

We conclude, by showing that it is also surjective. Consider any configuration $D \in \mathit{Conf}(\mathbb{P}(\mathsf{E}))$. Since D has no conflicts, its elements are pairwise compatible. Therefore, by coherence of the class of configurations, there exists $C \in \mathit{Conf}(E)$ such that $H \sqsubseteq C$ for all $H \in D$. Let $\mathit{fl}(D) = \bigcup D$. Then, for each $x \in \mathit{fl}(D)$ there exists a unique $H_x \in D$ such that $H_x \in \mathit{Hist}(x)$, since by Lemma 2.1(2) different histories of the same event are not compatible. Define the order $\leq_{\mathit{fl}(D)}$, for $x, y \in \mathit{fl}(D)$, by $x \leq_{\mathit{fl}(D)} y$ iff $x \in H_y$. It is easy to check that $\mathit{fl}(D) \sqsubseteq C$, and thus by prefix closedness of $\mathit{Conf}(\mathsf{E})$, we have $\mathit{fl}(D) \in \mathit{Conf}(\mathsf{E})$. It is now immediate to see that $hs(\mathit{fl}(D)) = D$, thus we conclude. □

The next lemma shows that every event structure E can be transformed into an hhp-bisimilar PESs $\mathbb{P}(\mathsf{E})$ which can be folded into E. For this reason we also say that E is unfolded to $\mathbb{P}(\mathsf{E})$.

Lemma 3.12 (unfolding event structures to PES's). *Let E be an event structure. Define a function $\phi_{\mathsf{E}} : \mathbb{P}(\mathsf{E}) \to \mathsf{E}$, for all $H \in \mathit{Hist}(\mathsf{E})$ by $\phi_{\mathsf{E}}(H) = x$ if $H \in \mathit{Hist}(x)$ for $x \in \mathsf{E}$. Then ϕ_{E} is a folding.*

Proof. The fact that ϕ_{E} is a morphism immediately follows from the observation that $\phi_{\mathsf{E}}(D) = \mathit{fl}(D)$. Then by Lemma 3.11, we have $D \simeq \phi_{\mathsf{E}}(D)$, as desired.

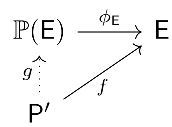
In order to conclude that it is a folding we show that given $D_1 \in \mathit{Conf}(\mathbb{P}(\mathsf{E}))$, if $\phi_{\mathsf{E}}(D_1) \xrightarrow{x} C_2$ then $D_1 \xrightarrow{H} D_2$ with $\phi_{\mathsf{E}}(D_2) = C_2$. Let $C_1 = \phi_{\mathsf{E}}(D_1)$ and assume $C_1 \xrightarrow{x} C_2$. By definition of transition (Definition 2.4), we have $C_1 \sqsubseteq C_2$. Let $H_x = C_2[x]$. By definition of $\mathbb{P}(\mathsf{E})$, the causes $\lfloor H_x \rfloor = \{H_x[y] \mid y \in H_x\}$. For all $y \in H_x \setminus \{x\}$, clearly $y \in C_1$. Moreover $H_x[y] = C_2[y] = C_1[y]$. Therefore, by Lemma 2.1(2), $H_x[y] \in D_1$. We thus conclude that

$$D_1 \xrightarrow{H_x} D_2$$

and moreover $\phi_{\mathsf{E}}(D_2) \simeq C_2$. For the last statement, the only thing to observe is that the image of the causes of H_x are exactly the causes of x. Indeed we have, for all $H \in D_2$, say $H \in \mathit{Hist}(y)$, that $H \sqsubseteq H_x$ iff $y \in H_x$ iff $y \leq_{C_2} x$, as desired. □

We next show that any morphism and any folding from a PES to an event structure E factorises uniquely through the PES $\mathbb{P}(\mathsf{E})$ associated with E (categorically, ϕ_{E} is cofree over E). This will be useful to relate foldings in E with foldings in $\mathbb{P}(\mathsf{E})$.

Lemma 3.13 (cofreeness of ϕ_{E})**.** *Let* E *be an event structure, let* P' *be a* PES *and let* $f : \mathsf{P}' \to \mathsf{E}$ *be an event structure morphism. Then there exists a unique morphism* $g : \mathsf{P}' \to \mathbb{P}(\mathsf{E})$ *such that* $f = \phi_{\mathsf{E}} \circ g$*.*



Moreover, when f *is a folding then so is* g*.*

Proof. The function g can be defined, for all $x' \in \mathsf{P}'$ as

$$g(x') = f(\lfloor x' \rfloor)$$

Note that this is a well-defined morphism. First observe that $g(x') \in \mathit{Hist}(\mathsf{E})$, hence it is an event in $\mathbb{P}(\mathsf{E})$. In fact, for all $x' \in \mathsf{P}'$, since f is a morphism and $\lfloor x' \rfloor \in \mathit{Conf}(\mathsf{P}')$, $f(\lfloor x' \rfloor) \in \mathit{Conf}(\mathsf{E})$, and $f(\lfloor x' \rfloor) \simeq \lfloor x' \rfloor$, therefore $g(x') = f(\lfloor x' \rfloor) = f(\lfloor x' \rfloor)[f(x')] \in \mathit{Hist}(\mathsf{E})$. Moreover, the reasoning above shows that $g(x') \in \mathit{Hist}(f(x'))$. Therefore, if $g(x') = g(y')$ then $f(x') = f(y')$. This fact, recalling that f is injective on configurations, implies that also g is. Finally, for all $C' \in \mathit{Conf}(\mathsf{P}')$, since f is a morphism, $f(C') \in \mathit{Conf}(\mathsf{E})$ and $f(C') \simeq C'$. Therefore its g-image is

$$\begin{aligned} g(C') &= \{g(x') \mid x' \in C'\} \\ &= \{f(\lfloor x' \rfloor) \mid x' \in C'\} \\ &= \{f(\lfloor x' \rfloor)[f(x')] \mid x' \in C'\} \quad \text{[Since morphisms preserve prefix order]} \\ &= \{f(C')[f(x')] \mid x' \in C'\} \\ &= \mathit{hs}(f(C')) \end{aligned}$$

Hence, by Lemma 3.11, $g(C') = \mathit{hs}(f(C')) \in \mathit{Conf}(\mathbb{P}(\mathsf{E}))$ and $\mathit{hs}(f(C')) \simeq C'$, as desired.

For the second part, assume that f is a folding and let us show that also g is. We use the characterisation in Lemma 3.3. Let $C'_1 \in \mathit{Conf}(\mathsf{P}')$ and assume that $g(C'_1) \xrightarrow{H} D_2$. Since ϕ_{E} is a morphism, this implies that $f(C'_1) = \phi_{\mathsf{E}}(g(C'_1)) \xrightarrow{\phi_{\mathsf{E}}(H)} \phi_{\mathsf{E}}(D_2)$. Since f is a folding, by Lemma 3.3, there exists a transition $C'_1 \xrightarrow{x'} C'_2$ such that $f(C'_2) = \phi_{\mathsf{E}}(C_2)$. Observe that this implies $f(x') = \phi_{\mathsf{E}}(H)$ and more generally $f(\lfloor x' \rfloor) = \phi_{\mathsf{E}}(\lfloor H \rfloor)$, but since $\phi_{\mathsf{E}}(\lfloor H \rfloor) = H$

$$f(\lfloor x' \rfloor) = H.$$

We only need to show that $g(C'_2) = D_2$. This is an immediate consequence of the fact that $g(C'_2) = g(C'_1) \cup \{g(x')\} = D_1 \cup \{H\} = D_2$, as desired. ☐

Remark 6. *Lemma 3.13 means that the category* **PES** *of prime event structures is a coreflective subcategory of* **ES***, i.e.,* $\mathbb{P} : \mathbf{ES} \to \mathbf{PES}$ *can be seen as a functor, right adjoint to the inclusion* $\mathbb{I} : \mathbf{PES} \to \mathbf{ES}$*. Moreover,* $\mathbb{P}$ *restricts to a functor on the subcategory of foldings,* $\mathbb{P} : \mathbf{ES_f} \to \mathbf{PES_f}$*, where an analogous result* *holds. This is in line with many classical results in the comparison of models of concurrency [36]. Intuitively, the existence of a coreflection means that for every event structure in* **ES** *there exists a* PES *which represents its best approximation in the category* **PES***, where the idea of approximation is formalised by the notion of morphism in the category.*

We conclude that all foldings between event structures arise from foldings of the associated PESs.

Proposition 3.14 (folding through PESs)**.** *Let* E, E' *be event structures. For all morphisms* $f : \mathsf{E} \to \mathsf{E}'$ *consider* $\mathbb{P}(f) : \mathbb{P}(\mathsf{E}) \to \mathbb{P}(\mathsf{E}')$ *defined by* $\mathbb{P}(f)(H) = f(H)$*. Then* f *is a folding iff* $\mathbb{P}(f)$ *is a folding.*

Proof. Let E, E' be event structures, let $f : \mathsf{E} \to \mathsf{E}'$ be a morphism and consider the commuting diagram

$$\begin{array}{ccc} \mathsf{E} & \xrightarrow{\;f\;} & \mathsf{E}' \\ {\scriptstyle \phi_{\mathsf{E}}}\uparrow & & \uparrow{\scriptstyle \phi_{\mathsf{E}'}} \\ \mathbb{P}(\mathsf{E}) & \xrightarrow[\mathbb{P}(f)]{} & \mathbb{P}(\mathsf{E}') \end{array}$$

If f is a folding then $f \circ \phi_{\mathsf{E}} : \mathbb{P}(\mathsf{E}) \to \mathsf{E}'$ is a composition of foldings and thus, by Lemma 2, it is a folding. In turn, by Lemma 3.13 this implies that $\mathbb{P}(f)$ is a folding.

Conversely, if $\mathbb{P}(f)$ is a folding, then $\phi_{\mathsf{E}} \circ \mathbb{P}(f) : \mathbb{P}(\mathsf{E}) \to \mathsf{E}'$ is a composition of foldings and thus, by Lemma 2, it is a folding. In turn, by Lemma 3.5 this implies that f is a folding. □

4. Foldings for Prime and Asymmetric Event Structures

In this section we study foldings on specific subclasses of poset event structures, providing suitable characterisations. Motivated by the fact that foldings on general poset event structures always arise from foldings of the corresponding canonical PESs we first and mainly focus on PESs. Then we discuss how this can be extended to asymmetric event structures (and only give a hint to flow and bundle event structures). We will see that while each PES admits a maximally folded version, for the other classes of event structures this does not happen in general.

4.1. Folding Prime Event Structures

Since foldings are special morphisms, we first provide a characterisation of PES morphisms.

Lemma 4.1 (PES morphisms)**.** *Let* P *and* P' *be* PES*s and let* $f : P \to P'$ *be a function on the underlying sets of events. Then* f *is a morphism iff for all* $x, y \in \mathsf{P}$

1. $\lambda'(f(x)) = \lambda(x)$*;*

2. *$f(\lfloor x \rfloor) = \lfloor f(x) \rfloor$; namely (a) for all $x' \in \mathsf{P}'$, if $x' \leq f(y)$ there exists $x \in \mathsf{P}$ such that $x \leq y$ and $f(x) = x'$ (b) if $x \leq y$ then $f(x) \leq f(y)$;*
3. *(a) if $f(x) = f(y)$ and $x \neq y$ then $x\#y$ and (b) if $f(x)\#f(y)$ then $x\#y$.*

Proof. First observe that PESs have global precedence (see Definition C.1) and $x \curvearrowright y$ iff $x \leq y$ or $x\#y$.

Now, assume that f is a morphism. Then property (1) holds by definition. Property (2) follows from the fact that $\lfloor x \rfloor \in \mathit{Conf}(\mathsf{P})$. Hence $f(\lfloor x \rfloor) \in \mathit{Conf}(\mathsf{P}')$ and $f(\lfloor x \rfloor) \simeq \lfloor x \rfloor$, which implies $f(\lfloor x \rfloor) = \lfloor f(x) \rfloor$.

Concerning condition (3b), observe that from Lemma C.2(1), instantiated with the notion of $\curvearrowright$ for PESs, we get

$$f(x) \leq f(y) \text{ or } f(x)\#f(y) \text{ implies } x \leq y \text{ or } x\#y.$$

In particular, if $f(x)\#f(y)$ then $x \leq y$ or $x\#y$ and, since conflict is symmetric, we also have $y \leq x$ or $y\#x$. It is now easy to see that only the second possibility $x\#y$ can hold true, which is the desired conclusion. Property (3a) immediately derives from Lemma C.2(2).

Conversely, assume that f satisfies conditions (1)-(3) above. Given a configuration $C \in \mathit{Conf}(\mathsf{P})$, by conditions (2a) and (3b), $f(C)$ is a configuration in P'. By condition (3a), f is injective on C. This, together with condition (2b), implies that $C \simeq f(C)$. □

Those in Lemma 4.1 are the standard conditions characterising (total) PES morphisms (see, e.g., [4]), with the addition of condition (2b) that is imposed to ensure that configurations are mapped to isomorphic configurations, as required by the notion of (strong) morphism (Definition 3.1).

We know that not all PES morphisms are foldings. We next identify some additional conditions characterising those morphisms which are foldings. The characterisation is later transferred to folding equivalences where it becomes simpler.

Theorem 4.2 (PES foldings)**.** *Let P and P' be PESs and let $f : \mathsf{P} \to \mathsf{P}'$ be a morphism. Then f is a folding if and only if it is surjective and for all $W \subseteq \mathsf{P}$, $x, y \in \mathsf{P}$, $y' \in \mathsf{P}'$*

1. *if $x\#^{\forall} f^{-1}(y')$ then $f(x)\#y'$;*
2. *if $f(x) = f(y)$, $\frown W$ and for all $w \in W$ $w \frown^{\exists} \{x, y\}$ then there exists $z \in \mathsf{P}$ such that $f(z) = f(x)$ and $\frown(W \cup \{z\})$.*

Proof. Let $f : \mathsf{P} \to \mathsf{P}'$ be a folding. Let us first observe that f is surjective. Take $x' \in \mathsf{P}'$. Since $\lfloor x' \rfloor \in \mathit{Conf}(\mathsf{P}')$, we have that $\emptyset \xrightarrow{\lfloor x' \rfloor} \lfloor x' \rfloor$. Since f is a folding, there must be $C \in \mathit{Conf}(\mathsf{P})$ such that $f(C) = \lfloor x' \rfloor$, and thus there is $x \in C$ such that $f(x) = x'$, as desired.

We next show that properties (1) and (2) hold.

1. We prove the contronominal, namely that if $f(x) \frown y'$ then there is $y \in \mathsf{P}$ such that $f(y) = y'$ and $x \frown y$. Assume that $f(x) \frown y'$. We distinguish two possibilities:

- If $y' \leq f(x)$ then, by Lemma 4.1(2a), there exists $y \leq x$ such that $f(y) = y'$. Hence $x \frown y$, as desired.
- Assume that, instead, $\neg(y' \leq f(x))$. Therefore, if we let $C' = \lfloor f(x) \rfloor \cup \lfloor y' \rfloor$ and $X' = C' \setminus \lfloor f(x) \rfloor$

$$\lfloor f(x) \rfloor \xrightarrow{X'} C' \tag{3}$$

By Lemma 4.1(2), we have that $f(\lfloor x \rfloor) = \lfloor f(x) \rfloor$. Therefore, since f is a folding, there must be a transition $\lfloor x \rfloor \xrightarrow{X} C$ with $f(C) = C'$. This means that there exists $y \in C$ such that $f(y) \in C'$ and, since $x \in C$, necessarily $x \frown y$, as desired.

2. Assume that $\frown W$, for all $w \in W$ $w \frown^{\exists} \{x, y\}$ and $f(x) = f(y)$. Define $C = \lfloor W \rfloor \in \mathit{Conf}(\mathsf{P})$. We distinguish two cases.
   - If $x \in C$ then we can simply take $z = x$, since clearly $\frown(W \cup \{x\})$.
   - Assume now that $x \notin C$. Clearly $f(x) \notin f(C)$. Moreover, $\frown(f(C) \cup \{f(x)\})$. In fact, by Lemma 4.1(3), if for some $u \in C$ it were $f(u) \# f(x) = f(y)$ there would exist $w \in W$ such that $f(w) \# f(x) = f(y)$. Hence we would have $w \# x$ and $w \# y$, contradicting the assumption $w \frown^{\exists} \{x, y\}$.

     Therefore $f(C) \xrightarrow{X'} f(C) \cup \lfloor f(x) \rfloor$ with $X' = f\lfloor f(x) \rfloor \setminus f(C)$. Since f is a folding, this implies that $C \xrightarrow{X} D$ with $f(D) = f(C) \cup \lfloor f(x) \rfloor$ and $D \simeq f(C) \cup \lfloor f(x) \rfloor$. Therefore there exists $z \in D$ such that $f(z) = f(x)$. Since $W \subseteq D$, we have that $\frown(W \cup \{z\})$, as desired.

For the converse implication, assume that f is a surjective morphisms satisfying conditions (1) and (2). We have to prove that it is a folding.

Let $C_1 \in \mathit{Conf}(\mathsf{P})$ and assume that $f(C_1) \xrightarrow{x'} C_2'$. If $C_1 = \emptyset$, take any $x \in \mathsf{P}$ such that $f(x) = x'$, which exists by surjectivity. By Lemma 4.1(2b) we have $f(\lfloor x \rfloor) = \lfloor x' \rfloor = \{x'\}$, and thus $\lfloor x \rfloor = \{x\}$. This means that $C_1 = \emptyset \xrightarrow{x} \{x\}$, and we conclude.
Otherwise, if $C_1 \neq \emptyset$, first observe that for all $y \in C_1$ since $f(y) \frown x'$, by condition (1), there exists some element $x_y \in \mathsf{P}$ such that $x_y \frown y$ and $f(x_y) = x$. Note that necessarily $\neg(x_y \leq y)$, otherwise, by Lemma 4.1(2b) we would have $x' = f(x_y) \leq f(y)$, which is not the case.
Since C_1 is finite and consistent, an inductive argument based on condition (2), allows us to derive the existence of x such that $f(x) = x'$ and $\frown(C_1 \cup \{x\})$. Moreover, as argued above for the x_y's, it is not the case that $x \leq y$ for some $y \in C_1$. Therefore there is a transition

$$C_1 \xrightarrow{X} C_1 \cup \lfloor x \rfloor$$

where $X = \lfloor x \rfloor \setminus C_1$.
We argue that $X = \{x\}$ and thus we conclude. In fact, assume that there is some $z \in X \setminus \{x\}$. Since f is a morphism $f(z) \leq f(x) = x'$. Now, since

there is the transition $f(C_1) \xrightarrow{x'}$, all causes of x' must be in $f(C_1)$. Note that, since f is a morphism, by Lemma 4.1(2), we have $\lfloor x' \rfloor = \lfloor f(x) \rfloor = f(\lfloor x \rfloor)$. Therefore, there must exist $z_1 \in C_1$ such that $f(z_1) = f(z)$. However, since $z, z_1 \in C_1 \cup (\lfloor x \rfloor \setminus \{x\})$ which is a configuration in $\mathit{Conf}(\mathsf{P})$, and f is injective on configurations, we get $z = z_1 \in C_1$, contradicting the hypothesis.

□

The notion of folding on PESs turns out to be closely related to that of abstraction homomorphism for PESs introduced in [29] for similar purposes. More precisely, abstraction homomorphisms can be characterised as those PES morphisms additionally satisfying condition (1) of Theorem 4.2, while they do not necessarily satisfy condition (2). Their more liberal definition is explained by the fact that they are designed to work on a subclass of structured PESs (see Appendix B for a detailed discussion).

We finally show what the conditions characterising foldings look like when transferred to equivalences.

Corollary 4.3 (folding equivalences for PESs)**.** *Let P be a PES and let $\equiv$ be an equivalence on P. Then $\equiv$ is a folding equivalence in $\mathit{FEq}(\mathsf{P})$ iff for all $x, y \in \mathsf{P}$, $x \neq y$, if $x \equiv y$ then*

1. $\lambda(x) = \lambda(y)$;
2. $[\lfloor x \rfloor]_\equiv = [\lfloor y \rfloor]_\equiv$;
3. $x \# y$.

Moreover, for all $x, y \in \mathsf{P}$, $W \subseteq \mathsf{P}$

4. *if $x \#^\forall [y]_\equiv$ then $[x]_\equiv \#^\forall [y]_\equiv$;*
5. *if $\frown W$ and for all $w \in W$, $w \frown^\exists [x]_\equiv$ then there exists $z \in [x]_\equiv$ such that $\frown(W \cup \{z\})$.*

Proof. Let P be a PESs and let $\equiv$ be a folding equivalence. This means that there exists a folding $f : \mathsf{P} \to \mathsf{P}'$ such that $\equiv$ and $\equiv_f$ coincide. By Lemma 3.4 we know that $\mathsf{P}_{/\equiv_f}$ is isomorphic to P'. Therefore using Lemma 4.1 and Theorem 4.2 we immediately get the validity of properties (1)-(4). Concerning property (5), we show that, more generally, if $\sim W$, $\{x_1, \ldots, x_n\} \subseteq [x]_\equiv$ and for all $w \in W$ $w \frown^\exists \{x_1, \ldots, x_n\}$ then there is $z \in [x]_\equiv$ such that $\frown (W \cup \{z\})$. The proof is by induction on n.

- if $n \leq 2$, we conclude by hypothesis.

- if $n > 2$, let us split $W = W' \cup W''$ in a way that for all $w' \in W'$ $w' \frown x_1$ and for all $w'' \in W''$ $w' \frown^\exists \{x_2, \ldots, x_n\}$. By inductive hypothesis, there is $z'' \in [x]_\equiv$ such that $\frown (W'' \cup \{z''\})$. Therefore we have that for all $w \in W = W' \cup W''$ $w \frown^\exists \{x_1, z''\}$. Now by hypothesis, we deduce the existence of z such that $f(x_1) = f(z)$ (hence $z \in [x]_\equiv$) such that $\frown (W \cup \{z\})$, as desired.

Conversely, assume that $\equiv$ satisfies properties (1)-(5) above. Define a PES P' as follows.

- $E' = E_{/\equiv}$;
- $[x]_\equiv \leq' [y]_\equiv$ if $[x]_\equiv \leq^\exists [y]_\equiv$;
- $[x]_\equiv \#' [y]_\equiv$ if $[x]_\equiv \#^\forall [y]_\equiv$
- $\lambda'([x]_\equiv) = \lambda(x)$.

Observe that P' is a well-defined PES. A simple key observation is that

$$[x]_\equiv \leq' [y]_\equiv \leq' [z]_\equiv \quad \Rightarrow \quad \exists x' \in [x]_\equiv.\ y' \in [y]_\equiv.\ z' \in [z]_\equiv.\ x \leq y \leq z \tag{4}$$

In fact, since $[y]_\equiv \leq' [z]_\equiv$, by definition we have the existence of $y' \in [y]_\equiv$ and $z' \in [z]_\equiv$ such that $y' \leq z'$. Moreover, since $[x]_\equiv \leq' [y]_\equiv$, by definition we have the existence of $x'' \in [x]_\equiv$ and $y'' \in [y]_\equiv$ such that $x'' \leq y''$. Since $y' \equiv y''$, by condition (2), $[y']_\equiv = [y'']_\equiv$. Hence from $x'' \leq y''$ we deduce the existence of $x' \leq y'$ with $x' \in [x]_\equiv$ as desired.

Using (4), we can immediately inherit the partial order properties of $\leq'$ and irreflexivity and hereditarity of $\#'$ from the analogous properties of $\#$.

If we define a function $f : \mathsf{P} \to \mathsf{P}'$ as $f(x) = [x]_\equiv$, it is now easy to show that it satisfies properties (1)-(3) in Lemma 4.1, and (1),(2) in Theorem 4.2, hence it is a folding and we conclude. □

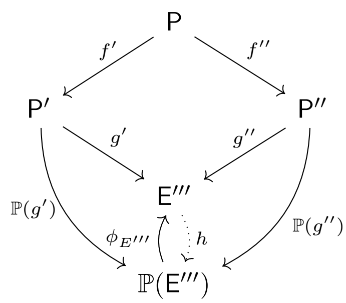
For instance, in Fig. 2, consider the equivalence $\equiv_{01}$ over P_0 such that $a_1 \equiv_{01} a_2$. This produces P_1 as quotient. This is not a folding equivalence since condition (4) fails: $a_1 \#^\forall [b_2]_{\equiv_{01}}$, but $\neg(a_2 \# b_2)$ and thus $\neg([a_1]_{\equiv_{01}} \#^\forall [b_2]_{\equiv_{01}})$. Instead, the equivalence $\equiv_{02}$ over P_0 such that $a_1 \equiv_{02} a_2$ and $b_1 \equiv_{02} b_2$, producing P_2 as quotient, satisfies all five conditions.

When PESs are finite, the result above suggests a possible way of identifying foldings: one can pair candidate events to be folded on the basis of conditions (1)-(3) and then try to extend the sets with condition (4)-(5) when possible. The procedure can be inefficient due to the global flavor of the conditions. This will be further discussed in the conclusions.

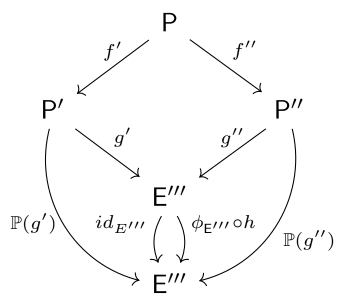
We know from Proposition 3.6 that all event structures admit a "maximally folded" version. We next observe that the same result holds in the class of PESs, i.e., that for each PES there is a uniquely determined minimal quotient.

Theorem 4.4 (joining foldings on PES's)**.** *Let $\mathsf{P}, \mathsf{P}', \mathsf{P}''$ be* PES*s and let $f' : \mathsf{P} \to \mathsf{P}'$, $f'' : \mathsf{P} \to \mathsf{P}''$ be foldings. Define E''' along with $g' : \mathsf{P}' \to \mathsf{E}'''$ and $g'' : \mathsf{P}'' \to \mathsf{E}'''$ as in Proposition 3.6. Then E''' is a* PES*. Therefore, any* PES *admits a unique minimal quotient which is a* PES*.*

Proof. The fact that $\mathrm{Pr}(g') : \mathsf{P}' \to \mathbb{P}(\mathsf{E}''')$ and $\mathbb{P}(g'') : \mathsf{P}'' \to \mathbb{P}(\mathsf{E}''')$ are foldings derive from Proposition 3.14. Now observe that, since In order to show that this actually provide a pushout in **PES**, consider two morphisms g_1' and g_2' as in the diagram below, such that $g_1' \circ f' = g_1'' \circ f''$:



Since E''' is a pushout and $\mathbb{P}(g') \circ f' = \mathbb{P}(g'') \circ f''$, there is a unique morphism $h : \mathsf{E}''' \to \mathbb{P}(\mathsf{E}''')$, making the diagram commute. Now, observe that $\phi_{\mathsf{E}'''} \circ h : \mathsf{E}''' \to \mathsf{E}'''$ can be used in the diagram below as mediating morphisms:



Now, since also the identity works as mediating morphisms we deduce that $h \circ \phi_{\mathsf{E}'''} = id_{\mathsf{E}'''}$, which implies that $\phi_{\mathsf{E}'''}$ is injective. Since it is a folding, it is also surjective, and therefore it is an isomorphism, as desired. □

Remark 7. *Theorem 4.4 can be also obtained as a consequence of the fact that the subcategory* $\mathbf{PES_f}$ *is a coreflective subcategory of* $\mathbf{ES_f}$ *and thus it is closed under pushouts as proved in [37, Corollary 1].*

4.2. Folding Asymmetric Event Structures

We know that foldings on all poset event structures arise from foldings on the corresponding canonical PESs. Still, for theoretical purposes and for efficiency reasons, a direct approach, not requiring the generation of the associated PES, can be of interest. Here we discuss the case of asymmetric event structures. This generalises the results in [27] that identify conditions which are only sufficient and apply to a subclass of foldings (the so-called called elementary foldings, merging a single set of events). Note also that, despite the fact that in this paper we work in a slightly different framework, we continue to have that, as observed in [27], AESs (and also FESs) do not admit a unique minimal quotient in general.

We first characterise morphisms in the sense of Definition 3.1 on AESs.

Lemma 4.5 (AES morphisms)**.** *Let* A *and* A' *be* AES*s and let* $f : A \to A'$ *be a function on the underlying sets of events. Then* f *is a morphism if and only if for all* $x, y \in \mathsf{A}$, $x \neq y$

1. $\lambda(f(x)) = \lambda(x)$*;*
2. $\lfloor f(x) \rfloor \subseteq f(\lfloor x \rfloor)$*;*
3. *(a) if* $f(x) \nearrow f(y)$ *then* $x \nearrow y$ *and (b) if* $x \nearrow y$ *and* $\neg(y \nearrow x)$ *then* $f(x) \nearrow f(y)$*;*
4. *if* $f(x) = f(y)$ *then* $x \nearrow y$*.*

Proof. Let $f : \mathsf{A} \to \mathsf{A}'$ be a morphism. Just observe that PESs have global precedence (see Definition C.1) and $x \curvearrowright y$ iff $x \nearrow y$. Condition (1) is obviously true. Property (2) follows by observing that, for all $x \in \mathsf{A}$, since $\lfloor x \rfloor \in \mathit{Conf}(\mathsf{A})$ and f is a morphism, then $f(\lfloor x \rfloor) \in \mathit{Conf}(\mathsf{A})$. Since configurations are causally closed we deduce that $\lfloor f(x) \rfloor \subseteq f(\lfloor x \rfloor)$. The validity of properties (3) and (4) is given directly by items (2) and (3) of Lemma C.2.

Conversely, assume that $f : \mathsf{A} \to \mathsf{A}'$ enjoys properties (1)-(4). Let $C \in \mathit{Conf}(\mathsf{A})$ be a configuration. Function f is injective on C since, otherwise, if there are $x, y \in C$ such that $f(x) = f(y)$ and $x \neq y$, we would get $x \nearrow y \nearrow x$, contradicting acyclicity of $\nearrow$ in C. Observe that $f(C)$ is a configuration. In fact, $\nearrow$ is acyclic in $f(C)$ since C is and, by (3a), cycles are reflected by f. In addition, $f(C)$ is causally closed by (2), since C is. Finally, note that $C \simeq f(C)$. In fact, for all $x, y \in C$, if $x \nearrow y$, since $\neg(y \nearrow x)$, by (3b), we get $f(x) \nearrow f(y)$. Conversely, if $f(x) \nearrow f(y)$ then $x \nearrow y$, by (3a). □

These are the standard conditions characterising (total) AES morphisms (see [6]), with the addition of (3b), needed in order to ensure that configurations are mapped to isomorphic configurations.

Proposition 4.6 (AES foldings)**.** *Let* A *and* A' *be* AES*s and let* $f : \mathsf{A} \to \mathsf{A}'$ *be a morphism. Then* f *is a folding if and only if it is surjective and for all* $X, Y \subseteq \mathsf{A}$, $x, y \in \mathsf{A}$ *with* $x \notin X$, $y \notin Y$, $y' \in \mathsf{A}'$

1. *if* $f^{-1}(y') \nearrow^{\forall} x$ *then* $y' \nearrow^{\exists} f(\lfloor x \rfloor)$*;*
2. *if* $\neg(x \nearrow^{\exists} X)$, $\neg(y \nearrow^{\exists} Y)$, $\frown(X \cup Y)$ *and* $f(x) = f(y)$ *then there exists* $z \in \mathsf{A}$ *such that* $f(z) = f(x)$ *and* $\neg(z \nearrow^{\exists} X \cup Y)$*.*
3. *given* $H \in \mathit{Hist}(x)$*, if* $\neg(H \nearrow^{\exists} X)$*, and* $H_1 \subsetneq H$ *such that* $f(H_1) \cup \{f(x)\} \in \mathit{Hist}(f(x))$ *there exists* x_1 *such that* $H_1 \cup \{x_1\} \in \mathit{Hist}(x_1)$ *and* $\neg(x_1 \nearrow^{\exists} X)$*.*

Proof. Let $f : \mathsf{A} \to \mathsf{A}'$ be a folding. Surjectivity of f can be proved exactly as in Theorem 4.2. We show that properties (1)-(3) hold.

1. We prove the contronominal, namely that if $\neg(y' \nearrow^{\exists} f(\lfloor x \rfloor))$ then there is $y \in \mathsf{A}$ such that $f(y) = y'$ and $\neg(y \nearrow x)$. Let $H = \lfloor x \rfloor \in \mathit{Conf}(\mathsf{A})$ and assume that $\neg(y' \nearrow^{\exists} f(H))$. Since f is a morphism $H' = f(H) \in \mathit{Hist}(f(x))$. Observe that we can safely assume that $y' \notin H'$. In fact, otherwise, since $\neg(y' \nearrow^{\exists} H')$, the only possibility would be $y' = f(x)$ and

thus we could take $y = x$ since $\neg(x \nearrow x)$, as desired. Using the fact that $\neg(y' \nearrow^{\exists} H')$ and $y \notin H'$, if we let $C' = H' \cup \lfloor y' \rfloor$ and $Y' = C' \setminus H'$

$$H' \xrightarrow{Y'} C' \tag{5}$$

Therefore, since f is a folding, there must be a transition $H \xrightarrow{X} C$ with $f(C) = C'$. This means that there exists $y \in X$ such that $f(y) = y'$ and since $H = \lfloor x \rfloor$, necessarily $\neg(y \nearrow x)$, as desired.

2. Assume that $x \notin X$, $y \notin Y$ $\neg(x \nearrow^{\exists} X)$, $\neg(y \nearrow^{\exists} Y)$, $\frown(X \cup Y)$ and $f(x) = f(y)$. Define $C = \lfloor X \cup Y \rfloor \in \mathit{Conf}(\mathsf{A})$. We show that $x \notin C$. In fact, $x \notin \lfloor X \rfloor$ since $x \notin X$ and $\neg(x \nearrow^{\exists} X)$, and, for analogous reasons, $y \notin \lfloor Y \rfloor$. Now, if $x = y$ we are done. Otherwise, we can prove that $x \notin \lfloor Y \rfloor$ and conclude. In fact, assume by contradiction that $x \in \lfloor Y \rfloor$, i.e., $x \leq w$ for some $w \in Y$. Since $f(x) = f(y)$ and $x \neq y$, we deduce, by Lemma 4.5(4), that $y \nearrow x$. Recalling $x \leq w$, by inheritance of asymmetric conflict, we get $y \nearrow^{\exists} Y$, contradicting the hypotheses.
Since $x \notin C$, we have $f(x) \notin f(C)$. Moreover, if we let $y' = f(x) = f(y)$, we have $\neg(y' \nearrow^{\exists} f(C))$. Otherwise, by Lemma 4.5(3a), we would deduce $x \nearrow^{\exists} X$ or $y \nearrow^{\exists} Y$, contradicting the hypotheses.
Therefore $f(C) \xrightarrow{X'} f(C) \cup \lfloor f(x) \rfloor$ with $X' = f\lfloor f(x) \rfloor \setminus f(C)$. Since f is a folding, this implies that $C \xrightarrow{X} D$ with $f(D) = f(C) \cup \lfloor f(x) \rfloor$ and $D \simeq f(C) \cup \lfloor f(x) \rfloor$. Therefore there exists $z \in D$ such that $f(z) = f(x)$. Therefore $\neg(z \nearrow^{\exists} C)$. Hence, recalling $C = \lfloor X \rfloor \cup \lfloor Y \rfloor$, we have $\neg(z \nearrow^{\exists} X \cup Y)$, as desired.
3. Take $H \in \mathit{Hist}(x)$ with $\neg(H \nearrow^{\exists} X)$ and $H_1 \subsetneq H$ such that $f(H_1) \cup \{f(x)\} \in \mathit{Hist}(f(x))$, hence $f(H_1) \xrightarrow{f(x)} f(H_1) \cup \{f(x)\}$. Consider $C = H_1 \cup \lfloor X \rfloor$. Since $H_1 \cup \{x\} \subseteq H$ and $\neg(H \nearrow^{\exists} X)$, we have $\neg(H_1 \cup \{x\} \nearrow^{\exists} \lfloor X \rfloor)$ and thus, by Lemma 4.5(3a), $\neg(f(H_1 \cup \{x\}) \nearrow^{\exists} f(\lfloor X \rfloor)$. Therefore $f(H_1 \cup \lfloor X \rfloor) = f(H_1) \cup f(\lfloor X \rfloor) \xrightarrow{f(x)} C_1'$, and since f is a folding $H_1 \cup \lfloor X \rfloor \xrightarrow{x_1} C_1$, with $f(x_1) = f(x)$ and clearly (given that the transition exists, $x_1 \nearrow^{\exists} X$, as desired.

For the converse implication, assume that f is a surjective morphism satisfying conditions (1)-(3). We have to prove that it is a folding.

Let $C_1 \in \mathit{Conf}(\mathsf{A})$ and assume that $f(C_1) \xrightarrow{x'} C_2'$. When $C_1 = \emptyset$ we argue as in Theorem 4.2. Otherwise, if $C_1 \neq \emptyset$, for all $y \in C_1$ it holds $\lfloor y \rfloor \subseteq C_1$ and thus $\neg(x' \nearrow^{\exists} f(\lfloor y \rfloor)$. Thus, by condition (1), there exists some element $x_y \in \mathsf{A}$ such that $f(x_y) = x'$ and $\neg(x_y \nearrow y)$. Note that necessarily $x_y \neq y$,

Since C_1 is finite and consistent, an inductive argument based on condition (2), allows us to derive the existence of x such that $f(x) = x'$ and $\neg(x \nearrow^{\exists} C_1)$. Therefore there is a transition

$$C_1 \xrightarrow{X} C_2$$

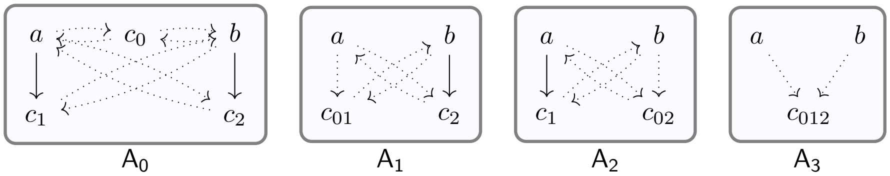


Figure 8: Asymmetric event structures do not admit a minimal quotient

where $C_2 = C_1 \cup \lfloor x \rfloor$ and $X = \lfloor x \rfloor \setminus C_1$.

Let $H = C_2[x]$. By definition of history, if $\neg(H \nearrow^{\exists} C_2 \setminus H)$. Let $H_1' = f(C_1)[x'] \setminus \{x'\}$ and let H_1 its f-counterimage in C_1. We have $H_1 \sqsubseteq H$, $x' = f(x) \notin f(H_1)$ and $f(H_1) \cup \{f(x)\} \in \mathit{Hist}(f(x))$. Then, by condition (3), there exists x_1 such that $H_1 \cup \{x_1\} \in \mathit{Hist}(x_1)$ and $\neg(x_1 \nearrow^{\exists} C_2 \setminus H)$, hence $\neg(x_1 \nearrow^{\exists} C_1 \setminus H_1)$. This implies $C_1 \xrightarrow{x_1} C_1 \cup \{x_1\}$, as desired. □

We already observed that working in the class of AESs we can obtain smaller quotients than in the class of PESs (see, e.g., the hhp-bisimilar structures P_2 in Fig. 2 and A_0 in Fig. 4). However, unsurprisingly, the folding criteria for AESs are less elegant and more complex than those for PESs. For a practical use, the reference to histories could cause a loss of efficiency. Moreover, the uniqueness of the minimal quotient is lost. Consider for instance the AESs in Fig. 8. It can be seen that $h_{01} : \mathsf{A}_0 \to \mathsf{A}_1$ is a folding where the events c_1, caused by a and c_0 in conflict with a, are merged in a single event c_{01} in asymmetric conflict with a. Similarly, $h_{02} : \mathsf{A}_0 \to \mathsf{A}_2$ is a folding obtained by merging c_0 and c_2. These are two minimal foldings that do not admit a join in the class of AESs. In fact, if we merge all three c-labelled events we obtain A_3, and it is easy to see that the function $h_{03} : \mathsf{A}_0 \to \mathsf{A}_3$ is not a folding. In fact, consider $\{a, b\} \in \mathit{Conf}(\mathsf{A}_0)$. Then $h_{03}(\{a, b\}) = \{a, b\} \xrightarrow{c_{012}}$, a transition that cannot be simulated in A_0. Indeed, it can be seen that the join of h_{01} and h_{02} is the event structure E in Fig. 1(right), which cannot be represented as an AES.

In passing, we note that also in the class of FESs and BESs the existence of minimal foldings is lost. In fact, consider the FESs in Fig. 5 (which can be also viewed as BESs). It can be easily seen that F_1 and F_2 are different minimal foldings of F_0. In particular, merging the three d-labelled events as in F_3 modifies the behaviour. In fact, in F_3, the event d_{012} is not enabled in $C = \{a\}$ since $c \prec d_{012}$ and no event in C is in conflict with c. Instead, in F_0, the event d_0 is clearly enabled from $\{a\}$.

Existence of a unique minimal folding could be possibly recovered by strengthening the notion of folding and, in particular, by requiring that foldings preserve and reflect histories. Note, however, that this would be against the spirit of our work where the notion of folding is not a choice. Rather, after having assumed hhp-bisimilarity as the reference behavioural equivalence, the notion of folding is essentially "determined" as a quotient (surjective function) that preserves the behaviour up to hhp-bisimilarity.

5. Conclusions

We studied the problem of minimisation for poset event structures, a class that encompasses many stable event structure models in the literature, assuming hereditary history-preserving bisimilarity as the reference behavioural equivalence. We showed that a uniquely determined minimal quotient always exists for poset event structures and also in the subclass of prime event structures, while this is not the case for various models extending prime event structures. We showed that foldings between general poset event structures arise from foldings of corresponding canonical prime event structures. Finally, we provided a characterisation of foldings of prime event structures, and discussed how this could be generalised to other classes, developing explicitly the case of asymmetric event structures.

We believe that, besides its original motivations from the setting of business process models and its foundational interest, this work can be of help in the study of minimisation, under a true concurrent equivalence, of operational models which can be mapped to event structures, like transition systems with independence or Petri nets.

As underlined throughout the paper, our theory of folding has many connections with the literature on event structures. The idea of "unfolding" more expressive models to prime algebraic domains and prime event structures has been studied by many authors (e.g., in [28, 3, 31, 32, 7]). The same can be said for the idea of refining a single action into a complex computation (see, e.g., [24] and references therein). Instead, the problem of characterising behaviour-preserving quotients of event structures has received less attention. We already commented on the relation with the notion of abstraction homomorphisms for PESs [29], which captures the idea of behaviour preserving abstraction in a subclass of structured PESs. In some cases, given a Petri net or an event structure a special transition system can be extracted, on which minimisation is performed. In particular, in [38] the authors propose an encoding of safe Petri nets into causal automata, in a way that preserves hp-bisimilarity. The causal automata can be transformed into a standard labelled transition system, which in turn can be minimised. However, in this way, the correspondence with the original events is lost.

The notion of behaviour preserving function has been given an elegant abstract characterisation in terms of open maps [25]. In the paper we mentioned the possibility, discussed in detail in Appendix A, of viewing our foldings as open maps and we observed that various results admit a categorical interpretation. This gives clear indications of the possibility of providing a general abstract view of the results in this paper, something which represents an interesting topic of future research.

The characterisation of foldings on prime (and asymmetric) event structures can be used as a basis to develop, at least in the case of finite structures, an algorithm for the definition of behaviour preserving quotients. The fact that conditions for folding refer to sets of events might make the minimisation procedure very inefficient. Determining suitable heuristics for the identification

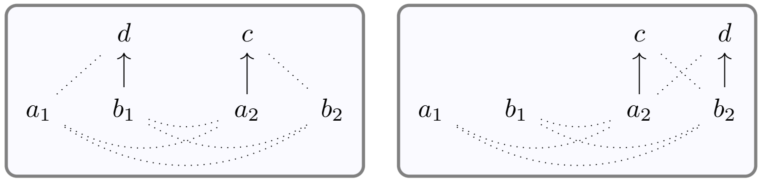


Figure 9: Two PESs involved in the absorption law

of folding sets and investigating the possibility of having more "local" conditions characterising foldings are interesting directions of future development.

Although not explicitly discussed in the paper, considering elementary foldings, i.e., foldings that just merge a single set of events, one can indeed determine some more efficient folding rules. This is essentially what is done for AESs and FESs in [27]. However, restricting to elementary foldings is limitative, since it can be seen that general foldings cannot be always decomposed in terms of elementary ones (e.g., it can be seen that in Fig. 2, the folding $f_{02} : \mathsf{P}_0 \to \mathsf{P}_2$ cannot be obtained as the composition of elementary foldings).

When dealing with possibly infinite event structures one could work on some finitary representation and try to devise reduction rules acting on the representation and inducing foldings on the corresponding event structure. Observe that working, e.g., on finite safe Petri nets, the minimisation procedure would be necessarily incomplete, given that hhp-bisimilarity is known to be undecidable [39].

A natural question concerns the possibility of extending the results in this paper to concurrent behavioural equivalences weaker than hereditary history-preserving bisimilarity. While reduction techniques of practical interest can be surely devised, we believe that the results in this paper, eminently the existence of a unique minimal quotient, can be hardly extended to other behavioural equivalences. An obvious candidate equivalence would be history-preserving bisimilarity, but the attempt would fall short. In fact, consider the PESs in Fig. 9, which are known to be history-preserving bisimilar but not hereditary history-preserving bisimilar [25]. One can easily realise that they are both minimal, i.e., no quotient preserves history-preserving bisimilarity. In fact two instances of both a and b are needed: one excluding and the other allowing for the execution of c and d, respectively. Technically, an important property that fails is the analogous of Lemma 3.8 for history-preserving bisimilarity, i.e., the possibility of viewing a history-preserving bisimilarity as an event structure. For even weaker notions of behavioural equivalence, like step or pomset bisimulation equivalence (see, e.g., [24]) the answer is less immediate. However, for equivalences which do not fully preserve the causal structure of computations it looks very difficult to be able to get a legal event structure model as a quotient.

In addition, we recall that hereditary history-preserving bisimilarity has been defined on other general models of concurrency, like configuration structures and higher-dimensional automata [40, 41]. Understanding whether the results in this paper can be generalised also to these settings is an interesting direction of future research.

A. Foldings as Open Maps

Here we observe that foldings, as defined in the paper, arise as open maps in the sense of [25]. We start by recalling the notion of open map.

Definition A.1 (open map)**.** Let $\mathbf{M}$ be a category and let $\mathbf{C}$ be a subcategory of $\mathbf{M}$. A morphism $f : M \to M'$ is $\mathbf{C}$-*open* if for all morphisms $e : C \to C'$ and commuting square

$$\begin{array}{ccc} C & \xrightarrow{\;c\;} & \mathsf{E} \\ {\scriptstyle e}\downarrow & \overset{c''}{\nearrow} & \downarrow{\scriptstyle f} \\ C' & \xrightarrow[c']{} & \mathsf{E}' \end{array}$$

there exists a morphism $c'' : C' \to \mathsf{E}$ such that the two triangles commute.

Let **Pom** denote the subcategory of **ES** having conflict-free PESs as objects and injective morphisms as arrows. Then we can show that foldings are **Pom**-open morphisms in **ES**, generalising to our setting a result proved for prime event structures in [25].

Lemma A.1 (foldings as open maps)**.** *Let* E, E' *be event structures and let* $f : \mathsf{E} \to \mathsf{E}'$ *be a morphism. Then* f *is a folding if and only if* f *is* **Pom**-*open.*

Proof. Let f be a folding. In order to prove that f is a **Pom**-open map, assume to have a commuting square as in Definition A.1. Since C is a conflict-free prime event structures, its set of events, ordered by causality, which abusing the notation, we still denote by C is a configuration. Since c is a morphism $c(C) \in \mathit{Conf}(\mathsf{E})$ and $c(C) \simeq C$, and thus $f(c(C)) \in \mathit{Conf}(\mathsf{E}')$ and $f(c(C)) \simeq C$. Similarly, $c'(C') \in \mathit{Conf}(\mathsf{E}')$ and $c'(C') \simeq C'$. Finally observe that $e(C) \sqsubseteq C'$. Thus $c'(e(C)) = f(c(C)) \sqsubseteq c'(C')$, meaning that $f(c(C)) \xrightarrow{X'} c'(C')$ for a suitable X'. By definition of folding, there must be a transition $c(C) \xrightarrow{X} D$ such that $f(D) = c'(C')$. Therefore, we can define $c'' : C' \to \mathsf{E}$ as follows: for all $x' \in C'$, let $c''(x')$ be the unique $y \in D$ such that $f(y) = c'(x')$.

Conversely, assume that f is an **Pom**-open map. We show that f satisfies the condition of Lemma 3.3. Let $C_1 \in \mathit{Conf}(\mathsf{E})$ and consider a transition $f(C_1) \xrightarrow{x'} C'_2$. If we view configurations C_1, C'_2 as pomsets, then we can build the following commuting square

$$\begin{array}{ccc} C_1 & \hookrightarrow & \mathsf{E} \\ {\scriptstyle f_{|C_1}}\downarrow & \overset{c''}{\nearrow} & \downarrow{\scriptstyle f} \\ C'_2 & \hookrightarrow & \mathsf{E}' \end{array}$$

By the fact that f is open, we get the morphism c'', and it is immediate to see that $C_1 \xrightarrow{x} c''(C'_2)$ is the desired transition that completes the proof. $\square$

The above characterisation of foldings as **Pom**-open maps and the fact that **PES** is a coreflective subcategory of **ES** (Lemma 3.13) can be exploited to derive some results in the paper. More precisely, Proposition 3.14 arises as an instance of [25, Lemma 6(iii)] and Lemma 3.12 of [25, Lemma 6(ii)].

B. Relating Foldings and Abstraction homorphisms

Abstraction homomorphisms have been introduced in [29] as a way of capturing behaviour preserving quotients of event structures. As mentioned in the main text, the mentioned work focuses on a subclass of "well-structured" event structures which can be obtained from the empty event structure by action-prefixing, non-deterministic choice $+$ and parallel composition $|$. This allows the author to have a more liberal notion of quotient. More precisely, we next show that abstraction homomorphisms can be characterised as those PES morphisms additionally satisying condition (1) of Theorem 4.2, while they do not necessarily satisfy condition (2).

In order to recall the notion of abstraction homomorphism, it is worth introducing some notation. Given a PES P and an event $x \in \mathsf{P}$ let us define $\lfloor x) = \lfloor x \rfloor \setminus \{x\}$, $\lceil x \rceil = \{y \mid y \in \mathsf{P} \ \wedge\ x < y\}$, and $conc(x) = \{y \mid y \in \mathsf{P} \ \wedge\ \neg(x \leq y \ \vee\ y \leq x \ \vee\ x\#y)\}$.

Definition B.1 (abstraction homomorphisms [29]). Let P, P' be PESs. An *abstraction morphism* is a function $f : \mathsf{P} \to \mathsf{P}'$ such that for all $x, y \in \mathsf{P}$

1. $\lambda'(f(x)) = \lambda(x)$;
2. $f(\lfloor x)) = \lfloor f(x))$;
3. $f(\lceil x \rceil) = \lceil f(x) \rceil$;
4. $f(conc(x)) = conc(f(x))$

Lemma B.1 (foldings vs abstraction homomorphisms). *Let* P, P' *be* PES *and let* $f : \mathsf{P} \to \mathsf{P}'$ *be a function. Then* f *is an abstraction morphism iff* f *is a* PES *morphism additionally satisying condition (1) of Theorem 4.2.*

Proof. Let f be an abstraction homomorphism. We first prove conditions (1)-(3) of Lemma 4.1. The first condition is already in Definition B.1. Condition (2), is immediately implied by Definition B.1(2) Concerning condition (3), let $x, y \in \mathsf{P}$ such that $f(x) = f(y)$ and $x \neq y$. Observe that we cannot have $x < y$, otherwise by Definition B.1(2), we would have $f(x) < f(y)$. Dually, it cannot be $y < x$. Moreover, it cannot be $x \in conc(y)$, otherwise Definition B.1(4) would be violated. Therefore, necessarily $x\#y$. The validity of condition (3b) is proved analogously.

We finally show that f satisfies also condition (1) of Theorem 4.2. Let $x \in \mathsf{P}$, $y \in \mathsf{P}'$ such that $\neg(f(x)\#y')$ and we show that $\neg(x\#y)$ for some $y \in \mathsf{P}$ such that $f(y) = y'$. We distinguish various possibilities:

- If $f(x) = y'$, we simply take $y = x$.

- If $y' < f(x)$, by Definition B.1(2) there exists $y \in \mathsf{P}$ with $y < x$ such that $f(y) = y'$, and we conclude.
- If $f(x) < y'$, by Definition B.1(3) there exists $y \in \mathsf{P}$ with $x < y$ such that $f(y) = y'$, and we conclude.
- If none of the above holds, necessarily $y' \in \mathit{conc}(f(x))x$, and thus by Definition B.1(4) there exists $y \in \mathsf{P}$ with $y \in \mathit{conc}(x)$ such that $f(y) = y'$, and we conclude.

Conversely, let f be a PES morphism additionally satisying condition (1) of Theorem 4.2. We prove that conditions (1)-(4) of Definition B.1 hold. As above, the first condition is already in Lemma 4.1. The second condition, namely $f(\lfloor x)) = \lfloor f(x))$ immediately follows from Lemma 4.1(2), i.e, $f(\lfloor x \rfloor) = \lfloor f(x) \rfloor$. In fact, we only need to observe that for all $y < x$, $f(y) \neq f(x)$, otherwise, by Lemma 4.1(3a) we would have $x \# y$.

Concerning (3), i.e., for $x \in \mathsf{P}$, $f(\lceil x \rceil) = \lceil f(x) \rceil$ let us prove separately the two inclusions.

- ($\subseteq$) Let $y' \in f(\lceil x \rceil)$, i.e., $y' = f(y)$ for some $y \in \lceil x \rceil$. Since $x < y$, by Lemma 4.1(2b), $f(x) < f(y)$ and thus $y' = f(y) \in \lceil f(x) \rceil$, as desired.
- ($\supseteq$) Let $y' \in \lceil f(x) \rceil$, i.e., $f(x) < y'$. Then, for all $y \in f^{-1}(y')$, since $f(x) < y' = f(y)$, by Lemma 4.1(2a), there is $z < y$ such that $f(z) = f(x)$. Hence either $z = x$ and thus $x < y$ or $z \neq x$, hence, by Lemma 4.1(3a), $x \# z$ and thus $x \# y$.

  It cannot be that $x \#^{\forall} f^{-1}(y')$, otherwise, by Theorem 4.2(1), we would have $x \# y$, which is not the case. Therefore there must exists $y \in f^{-1}(y')$ such that $x < y$. Therefore $y' = f(y) \in f(\lceil x \rceil)$.

Let us finally prove condition (4), i.e., for $x \in \mathsf{P}$, $f(\mathit{conc}(x)) = \mathit{conc}(f(x))$. Again, we prove separately the two inclusions.

- ($\subseteq$) Let $y' \in f(\mathit{conc}(x))$, i.e., $y' = f(y)$ for some $y \in \mathit{conc}(x)$. By Lemma 4.1(2b) and Lemma 4.1(3b), it must be $y' = f(y) \in \mathit{conc}(f(x))$, as desired.
- ($\supseteq$) Let $y' \in \mathit{conc}(f(x))$. Since $\neg(f(x) \# y')$, by Theorem 4.2(1), we deduce that $\neg(x \#^{\forall} f^{-1}(y'))$. Take any $y \in f^{-1}(y')$ such that $\neg(x \# y)$. Now observe that it cannot be $x < y$ or $y < x$, otherwise, by Lemma 4.1(2b) $f(x)$ and $y' = f(y)$ would be ordered in the same way, contradicting $y' \in \mathit{conc}(f(x))$. It cannot be $x = y$ either, otherwise $y' = f(y) = f(x)$, again contradicting $y' \in \mathit{conc}(f(x))$.

  Therefore, $y \in \mathit{conc}(x)$ and thus $y' = f(y) \in f(\mathit{conc}(x))$, as desired.

$\square$

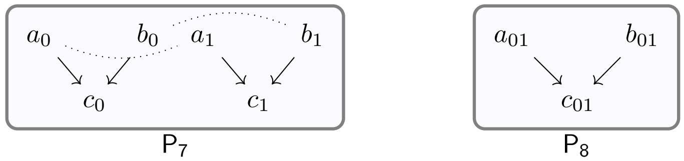


Figure B.10: Abstraction homomorphisms vs folding morphisms.

For instance, consider the PESs P_7 and P_8 in Fig. B.10. It can be seen that the obvious function $f_{78} : \mathsf{P}_7 \to \mathsf{P}_8$ is an abstraction homomorphism but not a folding. Indeed, consider the configuration $\{b_0, a_1\}$. Then the step $f_{78}(\{b_0, a_1\}) \xrightarrow{c_{01}} \{b_{01}, a_{01}, c_{01}\}$ cannot be simulated by $\{b_0, a_1\}$.

One can see that the PESs P_7 and P_8 are not in the subclass of well-structured PESs generated by the language considered in [29]. In fact none of the available operators can be used as the top operator: action-prefixing would produce a PES with a causally minimal event, while $+$ or $|$ would produce a PES whose events can partitioned into two blocks pairwise in conflict or concurrent, respectively.

C. Some Properties of Morphisms and Foldings

Here we define some relations between the events of an event structure, based on the way in which such events occur in configurations. They are used to prove general properties of morphisms and foldings of event structures, that then can be instantiated on specific subclasses.

Definition C.1 (precedence)**.** Let E be an event structure. The *precedence* as the relation $\curvearrowright \subseteq \mathsf{E} \times \mathsf{E}$, defined for $x, y \in \mathsf{E}$ by $x \curvearrowright y$ if for all $C \in \mathit{Conf}(\mathsf{E})$ such that $x, y \in C$ it holds $x <_C y$. We say that E has *global precedence* if for $x, y \in \mathsf{E}$, if $x, y \in C$ and $x <_C y$ then $x \curvearrowright y$.

In words, $x \curvearrowright y$ whenever in each computation where x, y occur necessarily x occurs before y. The precedence relation is useful also to define a notion of semantic conflict. Observe that for any configuration C the precedences expressed by $\curvearrowright$ are always respected by $\leq_C$, i,.e., $\curvearrowright_C^* \subseteq \leq_C$. When the event structure has global precedence, the precedence relation is sufficient to completely characterise the local order of configuration, i.e., for all configurations C it holds that $<_C = (\curvearrowright_{|C})^*$.

Closely connected, we can introduce a notion of semantic conflict.

Definition C.2 (conflict)**.** Let E be an event structure. The *conflict* is relation $\# \subseteq 2^E$, defined for a finite $X \subseteq E$ by $\#X$ if there is no $C \in \mathit{Conf}(\mathsf{E})$ such that $X \subseteq C$. When $\{x, y\}$ we often write $x\#y$.

We observe that conflict and precedence are strictly related. In particular, binary conflict can be characterised in terms of precedence.

Proposition C.1 (precedence vs conflict)**.** *Let E be an event structure. Then*

- *for $X \subseteq E$, if $\curvearrowright_{|X}$ is cyclic then $\#X$.*
- *for $x, y \in E$, we have $x\#y$ iff $x \prec y \prec x$.*

Proof.

- Let $X \subseteq E$. If $\curvearrowright_{|X}$ is cyclic, i.e., there are $x_1, \ldots, x_n \in X$ such that $x_1 \curvearrowright x_2 \curvearrowright \ldots x_n \curvearrowright x_1$ then the events $x_1, \ldots, x_n$ and thus X can never occur together in the same computation, i.e., there cannot be $C \in \mathit{Conf}(\mathsf{E})$ such that $X \subseteq C$. In fact, otherwise, we should have $\curvearrowright^*_{|C} \subseteq \leq_C$, contradicting the fact that $\leq_C$ is a partial order. In words, each of the events x_i should occur before the others, which is impossible.
- In particular, if $x\#y$ then x, y can never be in the same computation, hence trivially $x \prec y$ and $y \prec x$, and observe that also the converse holds.

□

Morphism on event structures can be shown to enjoy interesting properties with respect to the semantic relations.

Lemma C.2 (morphism properties)**.** *Let E, E' be event structures and let $f : \mathsf{E} \to \mathsf{E}'$ be a morphism. Then for all $x, y \in E$*

1. *if $f(x) \curvearrowright f(y)$ then $x \curvearrowright y$;*
2. *if $f(x) = f(y)$ then $x \curvearrowright y$, hence by duality $x\#y$.*

Moreover, if E, E' have global precedence, then

3. *if $x \curvearrowright y$ and $\neg(y \curvearrowright x)$ then $f(x) \curvearrowright f(y)$;*

Proof. Let $x, y \in E$

1. Assume $f(x) \curvearrowright f(y)$. Let $C \in \mathit{Conf}(\mathsf{E})$ be a configuration such that $x, y \in C$. Then $f(x), f(y) \in f(C)$ and $C \in \mathit{Conf}(\mathsf{E}')$. Since $f(x) \curvearrowright f(y)$ we have that $f(x) <_{f(C)} f(y)$ and thus, since f is a morphism, $x <_C y$. Since this holds for any configuration, we conclude $x \curvearrowright y$.
2. Assume $f(x) = f(y)$. Since f is injective on configurations, there cannot be $C \in \mathit{Conf}(\mathsf{E})$ such that $x, y \in C$. Therefore, trivially $x \curvearrowright y$ (and $y \curvearrowright x$, whence $x\#y$).
3. If E, E' have global precedence, f is a folding and $x \curvearrowright y$ and $\neg(y \curvearrowright x)$ then $\neg(x\#y)$ and thus there is some configuration $C \in \mathit{Conf}(\mathsf{E})$ such that $x, y \in C$. Since E has global precedence, $x \leq_C y$. Now $f(x), f(y) \in f(C)$ which is in $\mathit{Conf}(\mathsf{E}')$. Therefore $f(x) \leq_{f(C)} f(y)$. Again, since E' has global precedence, $f(x) \curvearrowright f(y)$, as desired.

□

References

[1] J. Hopcroft, R. Motwani, J. Ullman, Introduction to Automata Theory, Languages, and Computation, Addison-Wesley, 2006.

[2] A. Meyer, L. Stockmeyer, The equivalence problem for regular expressions with squaring requires exponential space, in: SWAT (FOCS), IEEE Computer Society, 1972, pp. 125–129.

[3] M. Nielsen, G. Plotkin, G. Winskel, Petri Nets, Event Structures and Domains, Part 1, Theoretical Computer Science 13 (1981) 85–108.

[4] G. Winskel, Event Structures, in: Petri Nets: Applications and Relationships to Other Models of Concurrency, Vol. 255 of LNCS, Springer, 1987, pp. 325–392.

[5] G. Winskel, Events, causality and symmetry, Computer Journal 54 (1) (2011) 42–57.

[6] P. Baldan, A. Corradini, U. Montanari, Contextual Petri nets, asymmetric event structures and processes, Information and Computation 171 (1) (2001) 1–49.

[7] G. Boudol, I. Castellani, Permutation of transitions: an event structure semantics for CCS and SCCS, in: Linear Time, Branching Time and Partial Order Semantics in Logics and Models for Concurrency, Vol. 354 of LNCS, Springer Verlag, 1988, pp. 411–427.

[8] G. Boudol, Flow Event Structures and Flow Nets, in: Semantics of System of Concurrent Processes, Vol. 469 of LNCS, Springer Verlag, 1990, pp. 62–95.

[9] R. Langerak, Bundle Event Structures: A Non-Interleaving Semantics for Lotos, in: 5th Intl. Conf. on Formal Description Techniques (FORTE'92), North-Holland, 1992, pp. 331–346.

[10] P. Baldan, A. Corradini, H. Ehrig, M. Löwe, U. Montanari, F. Rossi, Concurrent semantics of algebraic graph transformation systems, in: G. Rozenberg (Ed.), Handbook of Graph Grammars and Computing by Graph Transformation, Vol. III: Concurrency, World Scientific, 1999, pp. 107–187.

[11] P. Baldan, Modelling concurrent computations: from contextual Petri nets to graph grammars, Ph.D. thesis, University of Pisa (2000).

[12] G. Schied, On relating rewriting systems and graph grammars to event structures, in: H.-J. Schneider, H. Ehrig (Eds.), Dagstuhl Seminar 9301 on Graph Transformations in Computer Science, Vol. 776 of LNCS, Springer, 1994, pp. 326–340.

[13] G. Winskel, Event structure semantics for CCS and related languages, in: M. Nielsen, E. Schmidt (Eds.), Proceedings of ICALP'82, Vol. 140 of LNCS, Springer Verlag, 1982, pp. 561–576.

[14] R. Langerak, Transformation and Semantics for Lotos, Ph.D. thesis, Department of Computer Science, University of Twente (1992).

[15] D. Varacca, N. Yoshida, Typed event structures and the linear pi-calculus, Theoretical Computer Science 411 (19) (2010) 1949–1973.

[16] N. Busi, R. Gorrieri, A Petri Nets Semantics for π-calculus, in: Proceedings of CONCUR'95, Vol. 962 of LNCS, Springer Verlag, 1995, pp. 145–159.

[17] R. Bruni, H. Melgratti, U. Montanari, Event structure semantics for nominal calculi, in: C. Baier, H. Hermanns (Eds.), CONCUR 2006, Vol. 4137 of LNCS, Springer, 2006, pp. 295–309.

[18] S. Crafa, D. Varacca, N. Yoshida, Event structure semantics of parallel extrusion in the pi-calculus, in: L. Birkedal (Ed.), Proceedings of FoSSaCS'12, Vol. 7213 of LNCS, Springer, 2012, pp. 225–239.

[19] J. Pichon-Pharabod, P. Sewell, A concurrency semantics for relaxed atomics that permits optimisation and avoids thin-air executions, in: R. Bodík, R. Majumdar (Eds.), POPL 2016, ACM, 2016, pp. 622–633.

[20] A. Jeffrey, J. Riely, On thin air reads: Towards an event structures model of relaxed memory, in: M. Grohe, E. Koskinen, N. Shankar (Eds.), LICS 2016, ACM, 2016, pp. 759–767.

[21] S. Chakraborty, V. Vafeiadis, Grounding thin-air reads with event structures, PACMPL 3 (POPL) (2019) 70:1–70:28.

[22] M. Dumas, L. García-Bañuelos, Process mining reloaded: Event structures as a unified representation of process models and event logs, in: R. Devillers, A. Valmari (Eds.), Petri Nets 2015, Vol. 9115 of LNCS, Springer, 2015, pp. 33–48.

[23] M. Bednarczyk, Hereditary history preserving bisimulations or what is the power of the future perfect in program logics, Tech. rep., Polish Academy of Sciences (1991).

[24] R. van Glabbeek, U. Goltz, Refinement of actions and equivalence notions for concurrent systems, Acta Informatica 37 (4/5) (2001) 229–327.

[25] A. Joyal, M. Nielsen, G. Winskel, Bisimulation from open maps, Information and Computation 127 (2) (1996) 164–185.

[26] A. Armas-Cervantes, P. Baldan, M. Dumas, L. García-Bañuelos, Diagnosing behavioral differences between business process models: An approach based on event structures, Information Systems 56 (2016) 304–325.

[27] A. Armas-Cervantes, P. Baldan, L. García-Bañuelos, Reduction of event structures under history preserving bisimulation, Journal of Logical and Algebraic Methods in Programming 85 (6) (2016) 1110–1130.

[28] A. Rensink, Posets for configurations!, in: W. R. Cleaveland (Ed.), Proceedings of CONCUR'92, Vol. 630 of LNCS, Springer, 1992, pp. 269–285.

[29] I. Castellani, Bisimulations for concurrency, Ph.D. thesis, University of Edinburgh (1988).

[30] P. Baldan, A. Raffaetà, Minimisation of event structures, in: A. Chattopadhyay, P. Gastin (Eds.), Proceedings of FSTTCS'19, Vol. 150 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019, pp. 30:1–30:15.

[31] R. van Glabbeek, History preserving process graphs, Draft available at http://theory.stanford.edu/~rvg/abstracts.html#hppg (1996).

[32] R. van Glabbeek, G. Plotkin, Configuration structures, event structures and Petri nets, Theoretical Computer Science 410 (41) (2009) 4111–4159.

[33] G. Boudol, I. Castellani, Flow models of distributed computations: event structures and nets, Tech. Rep. 1482, INRIA, https://hal.inria.fr/inria-00075080 (1991).

[34] G. M. Pinna, A. Poigné, On the nature of events, in: Proceedings of MFCS'92, Vol. 629 of LNCS, Springer Verlag, 1992, pp. 430–441.

[35] G. M. Pinna, A. Poigné, On the nature of events: another perspective in concurrency, Theoretical Computer Science 138 (2) (1995) 425–454.

[36] G. Winskel, M. Nielsen, Models for concurrency, in: S. Abramsky, D. Gabbay, T. S. E. Maibaum (Eds.), Handbook of Logic in Computer Science. Volume 4: Semantic Modelling, Oxford University Press, 1995.

[37] H. Herrlich, G. Strecker, Coreflective subcategories, Transactions of the American Mathematical Society 157 (1971) 205–226.

[38] U. Montanari, M. Pistore, Minimal transition systems for history-preserving bisimulation, in: 14th Annual Symposium on Theoretical Aspects of Computer Science, Vol. 1200 of LNCS, Springer Verlag, 1997, pp. 413–425.

[39] M. Jurdzinski, M. Nielsen, J. Srba, Undecidability of domino games and hhp-bisimilarity, Information and Computation 184 (2) (2003) 343–368.

[40] R. J. van Glabbeek, On the expressiveness of higher dimensional automata, Theoretical Computer Science 356 (3) (2006) 265–290.

[41] P. Baldan, D. Gorla, T. Padoan, I. Salvo, Behavioural logics for configuration structures, Theoretical Computer Science 913 (2022) 94–112.