Representing Dependencies in Event Structures

G. Michele Pinna

To cite this version:

G. Michele Pinna. Representing Dependencies in Event Structures. 21th International Conference on Coordination Languages and Models (COORDINATION), Jun 2019, Kongens Lyngby, Denmark. pp.3-18, 10.1007/978-3-030-22397-7_1 . hal-02365495

HAL Id: hal-02365495

https://inria.hal.science/hal-02365495

Submitted on 15 Nov 2019

Representing Dependencies in Event Structures[⋆]

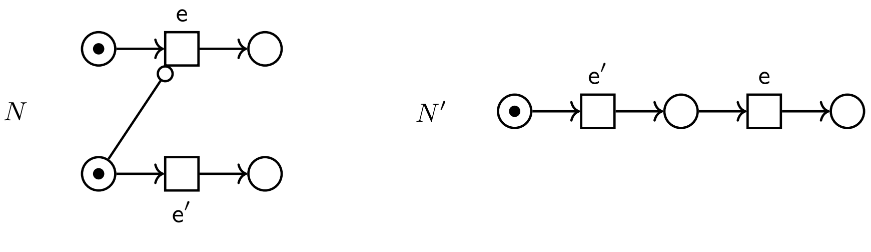
G. Michele Pinna

Dipartimento di Matematica e Informatica, Università di Cagliari, Cagliari, Italy

Abstract. Event Structures where the causality may change dynamically have been introduced recently. In this kind of Event Structures the changes in the set of the causes of an event are triggered by modifiers that may add or remove dependencies, thus making the happening of an event contextual. Still the focus is always on the dependencies of the event. In this paper we promote the idea that the *context* determined by the modifiers plays a major rôle, and the context itself determines not only the causes but also what causality should be. Modifiers are then used to understand when an event (or a set of events) can be added to a configuration, together with a set of events modeling dependencies, which will play a less important rôle. We show that most of the notions of Event Structure presented in literature can be translated into this new kind of Event Structure, preserving the main notion, namely the one of configuration.

1 Introduction

The notion of causality is an intriguing one. In the sequential case, the intuition behind it is almost trivial: if the activity e depends on the activity e', then to happen the activity e needs that e' has already happened. This is easily represented in Petri nets ([24]), the transition e' *produces* a token that is *consumed* by the transition e (the net N'). The dependency is testified by the observation that the activity e' always precedes the activity e. However this intuition does not reflect other possibilities. If we abandon the sequential case and move toward possibly loosely cooperating system the notion of causality become involved. Consider the case of a Petri net with inhibitor arcs ([13]) where the precondition of the



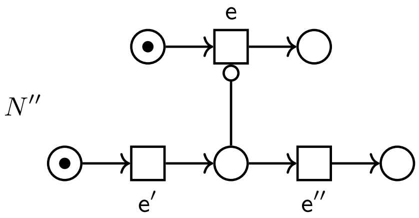
[⋆] Work partially funded by RAS (Regione Autonoma della Sardegna) - L.R. 7/2007 - Project SardCoin, CUP: F72F16003030002).

transition e' inhibits the transition e (the net N). The latter to happens needs that the transition e' happens first, and the *observation* testifies that the activity e needs that e' has already happened, though resources are not exchanged between e' and e. In both cases the observation that the event e' must happen first leads to state that e' precedes e and this can be well represented with a partial order relation among events.

This quite simple discussion suggests that the notion of causality may have many facets. In fact, if the dependencies are modeled just with a well founded partial order, inhibitor arcs can be used to model these dependencies, but the notion of partial order does not capture precisely the subtleties that are connected to the notion of causality.

To represent the semantics of concurrent systems the notion of *event structure* plays a prominent role. Event structures have been introduced in [18] and [27] and since then have been considered as a cornerstone. The idea is simple: the activities of a system are the *events* and their relationships are specified somehow, *e.g.* with a partial order modeling the *enabling* and a predicate expressing when activities are *coherent* or not. Starting from this idea many authors have faced the problem of adapting this notion to many different situations which have as a target the attempt to represent faithfully various situations. This has triggered many different approaches. In [11] and [12] *causal automata* are discussed, with the idea that the conditions under which an event may happen are specified by a suitable logic formula, in [10] and [9] it is argued that a partial order may be not enough or may be, in some situation, a too rigid notion, and this idea is used also in [21] and [22] where the notion of *event automata* is introduced, and it is used also in [20] where an enabling/disabling relation for event automata is discussed. Looking at the enabling relation, both *bundle event structures* ([15]) and *dual event structures* ([16]) provide a notion of enabling capturing *or*-causality (the former exclusive *or*-causality and the latter non exclusive *or*-causality). *Asymmetric event structures* ([6]) introduces a weaker notion of causality which models contextual arcs in Petri nets, or in the case *circular event structures* ([7]) the enabling notion is tailored to model also *circular dependencies*. In *flow event structures* ([8]) the partial order is required to hold only in configurations. Finally we mention the approaches aiming at modeling the possibility that the dependencies of an event may change either by dropping some of them or by adding new ones ([1]). This short and incomplete discussion (the event structures spectrum is rather broad) should point out the variety of approaches present in literature. It should be also observed that the majority of the approaches model causality with a relation that can be reduced to a partial order, hence causality is represented stating what are the events that should have happened before.

In this paper we introduce yet another notion of event structure. Triggered by recent works on *adding* or *subtracting* dependencies among events based on the fact that apparently unrelated events have happened ([1, 3]), we argue that rather than focussing on how to model these enrichment or/and impoverishment, it is much more natural to focus on the context on which an event takes place. In fact it is a context that can determine the proper dependencies that are applicable at the state where the event should take place and the context holds, and the context can also be used as well to forbid that the event is added to the state. This new relation resembles the one used in *inhibitor* event structures ([5]), but it differs in the way the contexts are determined. In the case of inhibitor event structures the presence of a certain event (the inhibiting context) was used to require that another one was present as well (representing the trigger able to remove the inhibition). Here the flavour is different as it is more prescriptive: it is required that exactly a set of events is present and if this happens then also another one should be present as well. It should be stressed that triggers and contexts may exchange their role. Consider again the two nets depicted before, we may have that in both cases the trigger is determined by the happening of the event e' and the context is the empty set, but we can consider as context the event e' and the trigger as the empty set. This simple relation, which we will call *context-dependency* relation, suffices to cover the various notions presented in literature. It is worth observing that determining the context and the triggers associated to it is quite similar to trying to understand the dependencies. Consider the net N'' below.



Here e may be added either to the empty set or to a set containing both e' and e''. The context containing e' only leads to require that the event e'' is present (in the spirit of the relation for inhibitor event structures), making e dependent on e''. However we could also have chosen to focus on contexts only and in this case the context containing just e' is ruled out among the contexts in which, together with some others dependencies, e may take place, and in this case the two contexts are $\emptyset$ and $\{\mathsf{e}', \mathsf{e}''\}$. As hinted above, it will turn out that the context plays a more relevant role with respect to the dependencies, as the context can be seen positively (it specifies under which conditions an event can be added, together with the dependency) or negatively (it specifies under which conditions an event can be added, and in this case the event cannot be added simply stipulating that it depends on itself).

In this paper we will focus on event structures where the change of state is always triggered by the happening of a single event, hence we will not consider steps (*i.e.* non empty and finite subsets of events), and where the states (configurations) are finite, though not always explicitly assumed. However the generalization to steps is straightforward.

Organization of the paper: In the next section we will introduce and discuss the new brand of event structure. In 3 we will review and briefly analyze some notions of event structures presented in literature, namely *prime* event structure ([27]), *relaxed prime* event structure and *dynamic causality* event structure ([1]), *inhibitor* event structure ([5]) and event structure for *resolvable conflicts* ([26]), and, in section 4, we show that the each event structure presented in section 3 can be translated into this new kind of event structure. We will recall also the notion *event automata* which will can used to compare the various notions of event structure. We will end the paper with some conclusions and we will give some hints for further developments.

Notation: Let A be a set, with let ρ we denote a sequence of elements belonging to A, and with ϵ we denote the empty sequence. With $\overline{\rho}$ we denote the set of elements of A appearing in ρ. Thus $\overline{\rho} = \emptyset$ if $\rho = \epsilon$ and $\overline{\mathsf{a}\rho'} = \{\mathsf{a}\} \cup \overline{\rho'}$ if $\rho = \mathsf{a}\rho'$. Given a sequence $\rho = \mathsf{a}_1 \cdots \mathsf{a}_n$ with $len(\rho)$ we denote its length, with ρ_0 we sometime denote sequence ϵ and, if $len(\rho) \geq 1$, for each $1 \leq i \leq len(\rho)$ with ρ_i we denote the sequence $\mathsf{a}_1 \cdots \mathsf{a}_i$. Let A be a set, with $\mathbf{2}^A$ we denote the subsets of A and with $\mathbf{2}^A_{fin}$ the finite subsets of A.

2 Context-Dependent Event Structure

We introduce yet another notion of event structure, which is the main contribution of the paper.

We start recalling what an *event* is and we introduce the notion of *configuration.* An event is an atomic individual action which is able to change the state of a system. Event structures in particular are intended to model *concurrent systems* by defining relationships among events such as *causality* and *conflict*, establishing the conditions on which a certain event can be added to a state. The *state* of a system modeled by an event structure is a *subset* of events (those happened so far), and this set of events is called *configuration.* States can be enriched by adding other information beside the one represented by the events that have determined the state, either adding information on the relationship among the various events in the state, *e.g.* adding dependencies among them (the state is then a partial order, [25]) or adding suitable information to the whole state.

We pursue this idea that the happening of an event depends on a set of modifiers (the *context*) and on a set of *real* dependencies, which are activated by the set of modifiers.

We recall that in this paper we will consider only *unlabelled* event structures. To simplify the presentation we retain the classic binary conflict relation. Given a subset $X \subseteq \mathsf{E}$ of events and a conflict relation $\#$, which is an irreflexive and symmetric relation, we say that X is *conflict free* iff $\forall \mathsf{e}, \mathsf{e}' \in X$ it holds that $\neg(\mathsf{e} \# \mathsf{e}')$.

Definition 1. *A* context-dependent event structure (CDES) *is a triple* $E = (\mathsf{E}, \#, \gg)$ *where*

- E *is a set of* events*,*
- $\# \subseteq \mathsf{E} \times \mathsf{E}$ *is an irreflexive and symmetric relation, called* conflict relation*, and*
- $\gg \subseteq \mathbf{2}^{\mathsf{A}} \times \mathsf{E}$*, where* $\mathsf{A} \subseteq \mathbf{2}^{\mathsf{E}}_{\mathit{fin}} \times \mathbf{2}^{\mathsf{E}}_{\mathit{fin}}$*, is a relation, called the* context-dependency *relation (*CD*-relation), which is such that for each* $\mathsf{Z} \gg \mathsf{e}$ *it holds that*
  - $\mathsf{Z} \neq \emptyset$*, and*
  - *for each* $(X, Y) \in \mathsf{Z}$ *it holds that* X *and* Y *are conflict-free.*

  Each element of the CD*-relation* $\gg$ *is called* entry*.*

The CD-relation models, for each event, which are the possible contexts in which the event may happen (the first component of each pair) and for each context which are the events that have to be occurred (the second component). We stipulate that dependencies and contexts are formed by non conflicting events, though this is not strictly needed, as the relation can model also conflicts. How this relation is used will become clear in the notion of enabling of an event. We have to determine, for each $\mathsf{Z} \gg \mathsf{e}$, which of the contexts X_i should be considered. To do so we define the *context* associated to each entry of the CD-relation. Given $\mathsf{Z} \gg \mathsf{e}$, where $\mathsf{Z} = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$, with $\textsc{Cxt}(\mathsf{Z})$ we denote the set of events $\bigcup_{i=1}^{|Z|} X_i$, and this is the one regarding $\mathsf{Z} \gg \mathsf{e}$.

Definition 2. *Let* $E = (\mathsf{E}, \#, \gg)$ *be a* CDES *and* $C \subseteq \mathsf{E}$ *be a subset of events. Then the event* $\mathsf{e} \notin C$ *is* enabled *at* C*, denoted with* $C[\mathsf{e}\rangle$*, if for each* $\mathsf{Z} \gg \mathsf{e}$*, with* $\mathsf{Z} = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$*, there is a pair* $(X_i, Y_i) \in \mathsf{Z}$ *such that* $\textsc{Cxt}(\mathsf{Z}) \cap C = X_i$ *and* $Y_i \subseteq C$*.*

Observe that requiring the non emptiness of the set Z in $\mathsf{Z} \gg \mathsf{e}$ guarantees that an event e may be enabled at some subset of events. The CD-relation could be used to express conflicts: $\mathsf{e} \# \mathsf{e}'$ could be modeled by adding $\{(\{\mathsf{e}\}, \{\mathsf{e}'\})\} \gg \mathsf{e}'$ and $\{(\{\mathsf{e}'\}, \{\mathsf{e}\})\} \gg \mathsf{e}$ to the $\gg$ relation, and the presence of just one of them would model the asymmetric conflict. The conflicts modeled in this way are *persistent*.

Definition 3. *Let* $E = (\mathsf{E}, \#, \gg)$ *be a* CDES*. Let* C *be a subset of* E*. We say that* C *is a* configuration *of the* CDES E *iff there exists a sequence of distinct events* $\rho = \mathsf{e}_1 \cdots \mathsf{e}_n \cdots$ *over* E *such that*

- $\overline{\rho} = C$*,*
- $\overline{\rho}$ *is conflict-free, and*
- $\forall 1 \leq i \leq len(\rho).\ \overline{\rho}_{i-1}[\mathsf{e}_i\rangle$*.*

Denoting with $\mathsf{Conf}_{\text{CDES}}(E)$ the set of configurations of a CDES, we introduce the relation among configurations. Given two configurations C and C' of a CDES such that $C \cup \{\mathsf{e}_{n+1}\} = C'$, we stipulate that $C \mapsto_{\text{CDES}} C'$ iff $C[\mathsf{e}_{n+1}\rangle$.

We illustrate this new kind of event structure with some examples.

Example 1. Consider three events a, b and c. All the events are singularly enabled but a and b are in conflict unless c has not happened (we will see later that this are called *resolvable* conflicts). Hence for the event a we stipulate

$$\{(\emptyset, \emptyset), (\{\mathsf{c}\}, \emptyset), (\{\mathsf{b}\}, \{\mathsf{c}\})\} \gg \mathsf{a}$$

that should be interpreted as follows: if the context is $\emptyset$ or $\{\mathsf{c}\}$ then a is enabled without any further condition (the Y are the empty set), if the context is $\{\mathsf{b}\}$ then also $\{\mathsf{c}\}$ should be present. The set $\textsc{Cxt}(\{(\emptyset, \emptyset), (\{\mathsf{c}\}, \emptyset), (\{\mathsf{b}\}, \{\mathsf{c}\})\})$ is $\{\mathsf{b}, \mathsf{c}\}$. Similarly, for the event b we stipulate

$$\{(\emptyset, \emptyset), (\{\mathsf{c}\}, \emptyset), (\{\mathsf{a}\}, \{\mathsf{c}\})\} \gg \mathsf{b}$$

which is justified as above and finally for the event c we stipulate

$$\{(\emptyset, \emptyset), (\{\mathsf{a}\}, \emptyset), (\{\mathsf{b}\}, \emptyset)\} \gg \mathsf{c}$$

namely any context allows to add the event.

Below we depict the configurations and how they are related.

$$\begin{tikzcd}
 & \{\mathsf{a}\} \arrow[r] & \{\mathsf{a}, \mathsf{c}\} \arrow[dr] & \\
\emptyset \arrow[ur] \arrow[r] \arrow[dr] & \{\mathsf{c}\} \arrow[ur] \arrow[dr] & & \{\mathsf{a}, \mathsf{b}, \mathsf{c}\} \\
 & \{\mathsf{b}\} \arrow[r] & \{\mathsf{b}, \mathsf{c}\} \arrow[ur] &
\end{tikzcd}$$

Example 2. Consider three events a, b and c, and assume that c depends on a unless the event b has occurred, and in this case this dependency is removed. Thus there is a classic causality between a and c, but it can dropped if b occurs. Clearly a and b are always enabled. The $\textsc{cd}$-relation is $\{(\emptyset, \emptyset)\} \gg \mathsf{a}$, $\{(\emptyset, \emptyset)\} \gg \mathsf{b}$ and $\{(\emptyset, \{\mathsf{a}\}), (\{\mathsf{b}\}, \emptyset)\} \gg \mathsf{c}$.

Example 3. Consider three events a, b and c, and assume that c depends on a just when the event b has occurred, and in this case this dependency is added, otherwise it may happen without Thus the classic causality relation between a and c is added if b occurs. Again a and b are always enabled. The $\textsc{cd}$-relation is $\{(\emptyset, \emptyset)\} \gg \mathsf{a}$, $\{(\emptyset, \emptyset)\} \gg \mathsf{b}$ and $\{(\emptyset, \emptyset), (\{\mathsf{b}\}, \{\mathsf{a}\})\} \gg \mathsf{c}$.

These examples should clarify how the $\textsc{cd}$-relation is used and its *expressivity.*

3 Event structures

We have introduced a new notion of event structure that we should confront with the others presented in literature (at least some of them). Therefore we review some of the various definitions of event structures.

Prime event structures: Prime event structures are one among the first proposed and the most widely studied ([27]), especially for the connections with *prime algebraic domains* and *causal nets*. The dependencies among events are modeled using a *partial order* relation, the incompatibility among events is modeled using a symmetric and irreflexive relation, the conflict relation, and it is required that the conflict relation is inherited along the partial order.

Definition 4. *A* prime event structure (PES) *is a triple* $P = (\mathsf{E}, \leq, \#)$*, where (a)* E *is a set of* events*, (b)* $\leq \subseteq \mathsf{E} \times \mathsf{E}$ *is a well founded* partial order *called* causality relation*, (c)* $\# \subseteq \mathsf{E} \times \mathsf{E}$ *is an irreflexive and symmetric relation, called* conflict relation*, such that* $\mathsf{e} \ \# \ \mathsf{e}' \leq \mathsf{e}'' \ \Rightarrow \ \mathsf{e} \ \# \ \mathsf{e}''$*, and (c)* $\leq \cap \# = \emptyset$*.*

Given an event $\mathsf{e} \in \mathsf{E}$, with $\lfloor e \rfloor$ we denote the set $\{e' \mid e' \leq e\}$, and the requirement that the partial order is well founded implies that for each $\mathsf{e} \in \mathsf{E}$, the set $\lfloor e \rfloor$ is finite. We say that C is a *configuration* of the PES P iff C is conflict free and for each $\mathsf{e} \in C$ it holds that $\lfloor \mathsf{e} \rfloor \subseteq C$. The set of configuration of a PES is denoted with $\mathsf{Conf}_{\text{PES}}(P)$. Clearly $(\mathsf{Conf}_{\text{PES}}(P), \subseteq)$ is a partial order. With $\mapsto_{\text{PES}}$ we denote the relation over $\mathsf{Conf}_{\text{PES}}(P) \times \mathsf{Conf}_{\text{PES}}(P)$ defined as $C \mapsto_{\text{PES}} C'$ iff $C \subset C'$ and $C' = C \cup \{\mathsf{e}\}$ for some $\mathsf{e} \in \mathsf{E}$.

Relaxed prime event structures: Some of the requirements of a PES, the one on the dependencies among events (here called enabling) and the ore regarding the conflicts among events (which does not need to be saturated), can be relaxed yielding a *relaxed* prime event structure ([1, 3]). In this definition the events that must be *present* in a state to allow the execution of another one are the events in a (finite) subset called *immediate causes* and often denoted with ic.

Definition 5. *A* relaxed *prime event structure (*rPES*) is a triple* $(\mathsf{E}, \rightarrow, \#)$*, where (a)* E *is a set of* events*, (b)* $\rightarrow \subseteq \mathsf{E} \times \mathsf{E}$ *is the* enabling relation *such that* $\forall \mathsf{e} \in \mathsf{E}$ *the set* $\text{ic}(\mathsf{e}) = \{\mathsf{e}' \mid \mathsf{e}' \rightarrow \mathsf{e}\}$ *is finite, and (c)* $\# \subseteq \mathsf{E} \times \mathsf{E}$ *is an irreflexive and symmetric* conflict relation*.*

The intuition is that the $\rightarrow$ relation plays the role of the causality relation and the conflict relation models conflicts among events, as before. The immediate causes can be seen as a mapping $\text{ic} \colon \mathsf{E} \rightarrow \mathbf{2}^{\mathsf{E}}_{\mathit{fin}}$. Let $T = (\mathsf{E}, \rightarrow, \#)$ be a rPES. Let C be a subset of E. We say that C is a *configuration* of the rPES T iff there exists a sequence of distinct events $\rho = \mathsf{e}_1 \cdots \mathsf{e}_n \cdots$ over E such that $\overline{\rho} = C$, $\overline{\rho}$ is *conflict free*, and for each $1 \leq i \leq \mathit{len}(\rho)$. $\text{ic}(\mathsf{e}_i) \subseteq \overline{\rho_{i-1}}$. The set of configuration of a rPES is denoted with $\mathsf{Conf}_{r\text{PES}}(T)$. In rPES the emphasis is put on the existence of an ordering in which the events are added to a configuration, and this will be valid for many of the kinds of event structures. $(\mathsf{Conf}_{r\text{PES}}(T), \subseteq)$ is a partial order. With $\mapsto_{r\text{PES}}$ we denote the relation over $\mathsf{Conf}_{r\text{PES}}(T) \times \mathsf{Conf}_{r\text{PES}}(T)$ defined as $C \mapsto_{r\text{PES}} C'$ iff $C \subset C'$ and $C' = C \cup \{\mathsf{e}\}$ for some $\mathsf{e} \in \mathsf{E}$.

A PES is also a rPES: the causality relation is the enabling relation and the conflict relation is the same one. e is added to a configuration C when its causes are in C and no conflict arises. Given a rPES $T = (\mathsf{E}, \rightarrow, \#)$, it is not difficult to see that $(\mathsf{E}, \rightarrow^*, \hat{\#})$ is a PES, where $\rightarrow^*$ is the reflexive and transitive closure of

$\rightarrow$ and $\hat{\#}$ is obtained by $\#$ stipulating that $\# \subseteq \hat{\#}$ and it is closed with respect to $\rightarrow^*$, *i.e.* if $\mathsf{e}\ \hat{\#}\ \mathsf{e}' \rightarrow^* \mathsf{e}''$ then $\mathsf{e}\ \hat{\#}\ \mathsf{e}''$. Indeed, the fact that $\rightarrow^*$ is a partial order is guaranteed by the fact that each event is executable, that $\rightarrow^*$ is well founded is implied by the finiteness of causes for each event $\mathsf{e} \in \mathsf{E}$ and $\hat{\#}$ is the semantic closure of $\#$: no new conflict is introduced.

Dynamic Causality Event Structures: We now review a notion of event structure where causality may change ([1, 3]). The idea is to enrich a rPES with two relations, one modeling the shrinking causality (some dependencies are dropped) and the other the growing causality (some dependencies are added). The shrinking and the growing causality relations are ternary relations stipulating that the happening of a specific event (the *modifier*) allows to drop or add a specific cause (the contribution) for another event (the *target*).

We illustrate these relations with the aid of a number of auxiliary subsets of events associated to these relations. Let E be a set of events. A *shrinking causality* relation is a ternary relation $\lhd\ \subseteq \mathsf{E} \times \mathsf{E} \times \mathsf{E}$, and the elements of this relation are denoted with $\mathsf{e}' \lhd [\mathsf{e} \rightarrow \mathsf{e}'']$. Given $\mathsf{e}' \lhd [\mathsf{e} \rightarrow \mathsf{e}'']$, e' is called *modifier*, e'' *target* and e *contribution*. $\mathsf{ShrMod}(\mathsf{e}'') = \{\mathsf{e}' \mid \mathsf{e}' \lhd [\mathsf{e} \rightarrow \mathsf{e}'']\}$ is the set of modifiers for a given target e'' and $\mathsf{Drop}(\mathsf{e}', \mathsf{e}'') = \{\mathsf{e} \mid \mathsf{e}' \lhd [\mathsf{e} \rightarrow \mathsf{e}'']\}$ is the set of contributions for a given modifier e' and a given target e''. Let H be a finite subset of E and let e be an event, we define the set $\mathrm{dc}(H, \mathsf{e}) = \bigcup_{\mathsf{e}' \in H \cap \mathsf{ShrMod}(\mathsf{e})} \mathsf{Drop}(\mathsf{e}', \mathsf{e})$ as the set of *dropped* causes with respect to H for the event e. A *growing causality* relation is a ternary relation $\blacktriangleright \subseteq \mathsf{E} \times \mathsf{E} \times \mathsf{E}$, and the elements of this relation are denoted as $\mathsf{e}' \blacktriangleright [\mathsf{e} \rightarrow \mathsf{e}'']$ Given $\mathsf{e}' \blacktriangleright [\mathsf{e} \rightarrow \mathsf{e}'']$, e' is called *modifier*, e'' *target* and e *contribution*. $\mathsf{GroMod}(\mathsf{e}'') = \{\mathsf{e}' \mid \mathsf{e}' \blacktriangleright [\mathsf{e} \rightarrow \mathsf{e}'']\}$ is the set of modifiers for a given target e'' and $\mathsf{Add}(\mathsf{e}', \mathsf{e}'') = \{\mathsf{e} \mid \mathsf{e}' \blacktriangleright [\mathsf{e} \rightarrow \mathsf{e}'']\}$ is the set of contributions for a given modifier e' and a given target e''. Let H be a finite subset of E and let e be an event, we define the set $\mathrm{ac}(H, \mathsf{e}) = \bigcup_{\mathsf{e}' \in H \cap \mathsf{GroMod}(\mathsf{e})} \mathsf{Add}(\mathsf{e}', \mathsf{e})$ as the set of *added* causes with respect to H for the event e. The two relation of shrinking and growing causality give the functions $\mathrm{dc}\colon \mathbf{2}^{\mathsf{E}}_{\mathit{fin}} \times \mathsf{E} \rightarrow \mathbf{2}^{\mathsf{E}}_{\mathit{fin}}$. and $\mathrm{ac}\colon \mathbf{2}^{\mathsf{E}}_{\mathit{fin}} \times \mathsf{E} \rightarrow \mathbf{2}^{\mathsf{E}}_{\mathit{fin}}$.

Definition 6. *A* dynamic causality event structure (DCES) *is a quintuple* $D = (\mathsf{E}, \rightarrow, \#, \lhd, \blacktriangleright)$*, where* $(\mathsf{E}, \rightarrow, \#)$ *is a* rPES*,* $\lhd\ \subseteq \mathsf{E} \times \mathsf{E} \times \mathsf{E}$ *is the* shrinking causality *relation,* $\blacktriangleright \subseteq \mathsf{E} \times \mathsf{E} \times \mathsf{E}$ *is the* growing causality *relation, and are such that for all* $\mathsf{e}, \mathsf{e}', \mathsf{e}'' \in \mathsf{E}$

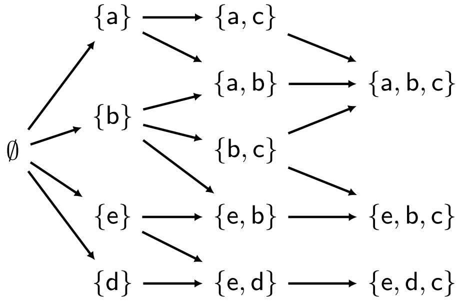
1. $\mathsf{e}' \lhd [\mathsf{e} \rightarrow \mathsf{e}''] \wedge \nexists \mathsf{e}''' \in \mathsf{E}.\ \mathsf{e}''' \blacktriangleright [\mathsf{e} \rightarrow \mathsf{e}''] \Longrightarrow \mathsf{e} \rightarrow \mathsf{e}''$*,*
2. $\mathsf{e}' \blacktriangleright [\mathsf{e} \rightarrow \mathsf{e}''] \wedge \nexists \mathsf{e}''' \in \mathsf{E}.\mathsf{e}''' \lhd [\mathsf{e} \rightarrow \mathsf{e}''] \Longrightarrow \neg(\mathsf{e} \rightarrow \mathsf{e}'')$*,*
3. $\mathsf{e}' \blacktriangleright [\mathsf{e} \rightarrow \mathsf{e}''] \Longrightarrow \neg(\mathsf{e}' \lhd [\mathsf{e} \rightarrow \mathsf{e}''])$*, and*
4. $\forall \mathsf{e}, \mathsf{e}' \in \mathsf{E}.\ \nexists \mathsf{e}'', \mathsf{e}''' \in \mathsf{E}.\ \mathsf{e}'' \lhd [\mathsf{e} \rightarrow \mathsf{e}']$ *and* $\mathsf{e}''' \blacktriangleright [\mathsf{e} \rightarrow \mathsf{e}']$.

For further comments on this definition we refer to [1] and [3]. It should be observed, however, that the definition we consider here is slightly less general of the one presented there, as we add a further condition, the last one, which is defined in [2] and does not allow that the same contribution can be added and removed by two different modifiers. These are called in [2] *single state dynamic causality event structures* and rule out the fact that some causality (or absence of) depends on the order of modifiers. Conditions 1 and 2 simply state that in the case of the shrinking relation the dependency should be present, and in the case of the growing the dependency should be absent; condition 3 says that if a dependency is added then it cannot be removed, or a removed dependency cannot be added, and the final condition express the fact that two modifiers, one growing and the other shrinking, cannot act on the same dependency. Clearly a DCES where $\lhd$ and $\blacktriangleright$ are empty is a rPES.

Let $D = (\mathsf{E}, \rightarrow, \#, \lhd, \blacktriangleright)$ be a DCES. Let C be a subset of E. We say that C is a *configuration* of the DCES iff there exists a sequence of distinct events $\rho = \mathsf{e}_1 \cdots \mathsf{e}_n$ over E such that (a) $\overline{\rho} = C$, (b) $\overline{\rho}$ is conflict-free, and (c) $\forall 1 \leq i \leq \mathit{len}(\rho).\ ((\mathrm{ic}(\mathsf{e}_i) \cup \mathrm{ac}(\overline{\rho_{i-1}}, \mathsf{e}_i)) \setminus \mathrm{dc}(\overline{\rho_{i-1}}, \mathsf{e}_i)) \subseteq \overline{\rho_{i-1}}$. The set of configuration of a DCES is denoted with $\mathsf{Conf}_{\mathrm{DCES}}(D)$.

With $\mapsto_{\mathrm{DCES}}$ we denote the relation over $\mathsf{Conf}_{\mathrm{DCES}}(D) \times \mathsf{Conf}_{\mathrm{DCES}}(D)$ defined as $C \mapsto_{\mathrm{DCES}} C'$ iff $C \subset C'$, $C' = C \cup \{\mathsf{e}\}$ for some $\mathsf{e} \in \mathsf{E}$ and $((\mathrm{ic}(\mathsf{e}) \cup \mathrm{ac}(C, \mathsf{e})) \setminus \mathrm{dc}(C, \mathsf{e})) \subseteq C$.

Example 4. Consider the set of events $\{\mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{d}, \mathsf{e}\}$, with $\mathsf{b} \rightarrow \mathsf{c}$, $\mathsf{a} \lhd [\mathsf{b} \rightarrow \mathsf{c}]$, $\mathsf{d} \blacktriangleright [\mathsf{e} \rightarrow \mathsf{c}]$, $\mathsf{a} \# \mathsf{e}$ and $\mathsf{d} \# \mathsf{b}$. a and d are the modifiers for the target c, the happening of a has the effect that the cause b may be dropped, and the one of d that the cause e should be added for c. If the prefix of the trace is bc (the target c is executed before of one of its modifiers a and d) then the final part of the trace is any either a or e, and as $\mathsf{d} \# \mathsf{b}$ we have that d cannot be added. If the modifier a is executed before c then we have the traces ac (as the immediate cause b of c is dropped by a) followed by b or d, and if the modifier d is executed, then before adding c, we need e (the modifier d add the immediate cause e for c), and in this case we cannot add b for sure as it is in conflict with d or a as it is in conflict with e. If both modifiers a and d happen, then the event c is permanently disabled, as it needs the contribution e (growing cause) which is in conflict with a. Below are shown the configurations of this DCES and the $\mapsto_{\mathrm{DCES}}$ relation.



A shrinking event structure (SES) is a DCES where the $\blacktriangleright$ relation is empty and a growing event structure (GES) is a DCES where the $\lhd$ relation is empty.

Inhibitor event structures: Inhibitor event structure ([5]) are equipped with a relation $\vdash\!\circ \ \subseteq \mathbf{2}^{\mathsf{E}}_{1} \times \mathsf{E} \times \mathbf{2}^{\mathsf{E}}_{\mathit{fin}}$ allowing to model conflicts (even asymmetric) as well as temporary inhibitions. With $\mathbf{2}^{\mathsf{E}}_{1}$ we denote the subsets of events with cardinality at most one (the empty set or singletons). The intuition behind this relation is the following: given $\vdash\!\circ(a, \mathsf{e}, A)$, the event e is enabled at a configuration is whenever the configuration contains the set a, then its intersection with A is non empty. Hence the event in a non empty a *inhibits* the happening of e unless some event in A has happened as well. We stipulate that given $\vdash\!\circ(a, \mathsf{e}, A)$ the events in A are pairwise conflicting (denoted with $\#(A)$). Two events e and e' are in conflict if $\vdash\!\circ(\{\mathsf{e}'\}, \mathsf{e}, \emptyset)$ and $\vdash\!\circ(\{\mathsf{e}\}, \mathsf{e}', \emptyset)$. An *or*-causality relation $<$ is definable stipulating that $A < \mathsf{e}$ if $\vdash\!\circ(\emptyset, \mathsf{e}, A)$, and that if $A < \mathsf{e}$ and $B < \mathsf{e}'$ for some $\mathsf{e}' \in A$ then also $B < \mathsf{e}$. This relation should be interpreted as follows: $A < \mathsf{e}$ means that if e is present, then also an event in A should be present.

Definition 7. *An* inhibitor event structure (IES) *is a pair* $I = (\mathsf{E}, \vdash\!\circ)$, *where* E *is a set of* events *and* $\vdash\!\circ \ \subseteq \mathbf{2}^{\mathsf{E}}_{1} \times \mathsf{E} \times \mathbf{2}^{\mathsf{E}}_{\mathit{fin}}$ *is a relation such that for each* $\vdash\!\circ(a, \mathsf{e}, A)$ *it holds that* $\#(A)$ *and* $a \cup A \neq \emptyset$.

We briefly recall the intuition: consider an event e and a triple in the $\vdash\!\circ$ relation $\vdash\!\circ(a, \mathsf{e}, A)$. Then e can be added provided that if the event in a is present also one in A should be present. Let $I = (\mathsf{E}, \vdash\!\circ)$ be an IES. Let C be a subset of E. We say that C is a *configuration* of the IES I iff there exists a sequence of distinct events $\rho = \mathsf{e}_1 \cdots \mathsf{e}_n \cdots$ over E such that $\overline{\rho} = C$ and for each $i \leq n$, for each $\vdash\!\circ(a, \mathsf{e}_i, A)$, it holds that $a \subseteq \overline{\rho_{i-1}} \ \Rightarrow\ \overline{\rho_{i-1}} \cap A \neq \emptyset$. The set of configuration of a IES is denoted with $\mathsf{Conf}_{\text{IES}}(I)$. With $\mapsto_{\text{IES}}$ we denote the relation over $\mathsf{Conf}_{\text{IES}}(I) \times \mathsf{Conf}_{\text{IES}}(D)$ defined as $C \mapsto_{\text{IES}} C'$ iff $C \subset C'$ and $C' = C \cup \{\mathsf{e}\}$ for some $\mathsf{e} \in \mathsf{E}$.

Example 5. Consider three events a, b and c, $\vdash\!\circ(\{\mathsf{a}\}, \mathsf{c}, \{\mathsf{b}\})$ and $\vdash\!\circ(\emptyset, \mathsf{b}, \{\mathsf{a}\})$. The maximal event traces are cab and abc. The event c is inhibited when the event a has occurred unless the event b has occurred as well. The configurations are $\emptyset$, $\{\mathsf{a}\}$, $\{\mathsf{c}\}$, $\{\mathsf{a}, \mathsf{b}\}$, $\{\mathsf{a}, \mathsf{c}\}$ and $\{\mathsf{a}, \mathsf{b}, \mathsf{c}\}$ and are reached as follows: $\emptyset \mapsto_{\text{IES}} \{\mathsf{a}\}$, $\emptyset \mapsto_{\text{IES}} \{\mathsf{c}\}$, $\{\mathsf{a}\} \mapsto_{\text{IES}} \{\mathsf{a}, \mathsf{b}\}$, $\{\mathsf{c}\} \mapsto_{\text{IES}} \{\mathsf{a}, \mathsf{c}\}$, $\{\mathsf{a}, \mathsf{b}\} \mapsto_{\text{IES}} \{\mathsf{a}, \mathsf{b}, \mathsf{c}\}$ and $\{\mathsf{a}, \mathsf{c}\} \mapsto_{\text{IES}} \{\mathsf{a}, \mathsf{b}, \mathsf{c}\}$.

Event structures with resolvable conflicts: We finally recall the notion of event structure with resolvable conflicts ([26]).

Definition 8. *An* event structure with resolvable conflicts *(*RCES*) is the pair* $R = (\mathsf{E}, \vdash)$ *where* E *is a set of events and* $\vdash \subseteq \mathbf{2}^{\mathsf{E}} \times \mathbf{2}^{\mathsf{E}}$ *is the* enabling *relation.*

No restriction is posed on the enabling relation. The intuition is that stipulating $X \vdash Y$ one state that for all the events in Y to occur, also the events in the set X should have occurred first.

The *single event* transition relation $\rightsquigarrow \subseteq \mathbf{2}^{\mathsf{E}} \times \mathbf{2}^{\mathsf{E}}$ of a RCES $R = (\mathsf{E}, \vdash)$ is given by $X \rightsquigarrow Y \Leftrightarrow (X \subseteq Y \wedge |Y \setminus X| \leq 1 \wedge \forall Z \subseteq Y.\ \exists W \subseteq X.\ W \vdash Z)$. With this notion it is possible to define what a configuration is: it is a subset X of events such that $X \rightsquigarrow X$. The requirement that $X \rightsquigarrow X$ implies that each subset of events is enabled in the configuration. Let $R = (\mathsf{E}, \vdash)$ be a RCES. Let C be a subset of E. We say that C is a *configuration* of the IES I iff there exists a sequence of distinct events $\rho = \mathsf{e}_1 \cdots \mathsf{e}_n \cdots$ over E such that for each $1 \leq i \leq len(\rho)$ it holds that $\overline{\rho_{i-1}}$ and $\overline{\rho_i}$ are configurations, and $\overline{\rho_{i-1}} \rightsquigarrow \overline{\rho_i}$. The set of configuration of a RCES is denoted with $\mathsf{Conf}_{\text{RCES}}(R)$.

Given two configurations C and C' of a RCES, such that $C \cup \{\mathsf{e}_{n+1}\} = C'$ and $C \rightsquigarrow C$, we stipulate again that $\overline{\rho} \mapsto_{\text{DCES}} \overline{\rho'}$, defining a relation over $\mathsf{Conf}_{\text{RCES}}(R) \times \mathsf{Conf}_{\text{RCES}}(R)$. Observe that the enabling relation $\vdash$ is used not only to state under which condition an event may happen but also to stipulate when an event is *deducible* from a set of events, justifying also the deduction symbol used for this relation. Observe also that $\mapsto_{\text{RCES}}$ is essentially $\rightsquigarrow$.

Example 6. Consider three events a, b and c, and $\emptyset \vdash X$ where $X \subseteq \{\mathsf{a}, \mathsf{b}, \mathsf{c}\}$ with $X \neq \{\mathsf{a}, \mathsf{b}\}$ and $\{\mathsf{c}\} \vdash \{\mathsf{a}, \mathsf{b}\}$. The intuition is that all the events are singularly enabled but a and b are in conflict unless c has not happened. In fact $\{\mathsf{a}, \mathsf{b}\}$ is not a configuration as taking $\{\mathsf{a}, \mathsf{b}\}$ as the $Z \subseteq \{\mathsf{a}, \mathsf{b}\}$ of the notion of single event transition relation, there is no subset of $\{\mathsf{a}, \mathsf{b}\}$ enabling these two events.

The configurations and how they are reached are those of the Example 1.

4 Embedding and comparing Event Structures

We now show that each of the event structure we have seen so far can be seen as a CDES, and also how to compare them. For the sake of simplicity, we will consider event structures where each event e is *executable*, namely that there is at least a configuration containing it.

Comparing Event Structures: We start by devising how we can compare two event structures of any kind. The intuition is obvious: two event structures are equivalent iff they have the same configurations and the $\mapsto$ relations defined on configurations coincide. We recall the notion of *event automaton* ([22]).

Definition 9. *Let* E *be a set of* events. *An* event automaton *over* E *(*ea*) is the tuple* $\mathcal{E} = \langle \mathsf{E}, \mathsf{S}, \mapsto, s_0 \rangle$ *such that*

- $\mathsf{S} \subseteq \mathbf{2}^{\mathsf{E}}$*, and*
- $\mapsto \subseteq \mathsf{S} \times \mathsf{S}$ *is such that* $s \mapsto s'$ *implies that* $s \subset s'$.

$s_0 \in \mathsf{S}$ *is the initial state.*

Event automata can easily express configurations of any kind of event structure, provided that for each kind a way to reach a configuration from another is given. The kind of event structure is ranged over by $\mu, \mu' \in \{$PES, rPES, DCES, IES, RCES, CDES$\}$.

Theorem 1. *Let* X *be an event structure of kind* μ *over the set of events* E*. Then* $\mathcal{G}_\mu(X) = \langle \mathsf{E}, \mathsf{Conf}_\mu(X), \mapsto_\mu, \emptyset \rangle$ *is an event automaton.*

Using event automata we can decide when two event structures are equivalent.

Definition 10. *Let X and Y be event structures over the same set of events E of kind μ and μ' respectively. We say that X and Y are* equivalent*, denoted with $X \equiv Y$, iff $\mathcal{G}_{\mu}(X) = \mathcal{G}_{\mu'}(Y)$.*

The expressivity is explicitly studied in [1] and [3]. Informally a kind of event structure is more expressive with respect to another, when there is a configuration of the former that cannot be a configuration of the latter, whatever is done with the various relations among events. Incomparable means that neither one is more expressive than the other or the vice versa. We shortly summarize part of these findings, when considering finite configurations. PES and rPES are equally expressive, whereas SES and GES are strictly more expressive than rPES, and are incomparable one with respect to the other. These two are both less expressive than DCES and RCES, which are incomparable. The relative expressivity of other kinds of event structure has not been investigated.

Embedding event structures into CDES*:* We prove now a more general result, namely that given any *event automaton* $\mathcal{E}$, which is obtained by the configurations of any kind of event structure, it is possible to obtain a CDES whose configurations are precisely the ones of the event automaton $\mathcal{E}$. We start identifying, in an ea, the events that are in *conflict.* The conflict relation we obtain is a *semantic* conflict relation: two events are in conflict iff they never appear together in a state.

Definition 11. *Let $\mathcal{E} = \langle \mathsf{E}, \mathsf{S}, \mapsto, s_0 \rangle$ be an ea. We define a symmetric and irreflexive conflict relation $\#_{\mathsf{ea}}$ as follows: $\mathsf{e} \ \#_{\mathsf{ea}} \ \mathsf{e}'$ iff for each $s \in \mathsf{S}$. $\{\mathsf{e}, \mathsf{e}'\} \not\subseteq s$.*

In order to obtain the CD-relation we need some further definitions. Fixed an event e, the first one identifies the states where this event can be added, and the second one identifies the states where the event cannot be added.

Definition 12. *Let $\mathcal{E} = \langle \mathsf{E}, \mathsf{S}, \mapsto, s_0 \rangle$ be an ea. To each event $\mathsf{e} \in \mathsf{E}$ we associate the subset of events $\{s \in \mathsf{S} \mid s \cup \{\mathsf{e}\} \in \mathsf{S} \ \wedge \ s \mapsto s \cup \{\mathsf{e}\}\}$, which we denote with $\mathcal{C}(\mathcal{E}, \mathsf{e})$.*

Definition 13. *Let $\mathcal{E} = \langle \mathsf{E}, \mathsf{S}, \mapsto, s_0 \rangle$ be an ea. To each event $\mathsf{e} \in \mathsf{E}$ we associate the set of configuration $\{s \in \mathsf{S} \mid s \cup \{\mathsf{e}\} \not\in \mathsf{S}\}$, which we denote with $\mathcal{I}(\mathcal{E}, \mathsf{e})$.*

Definition 12 characterizes when an event is enabled giving the *allowing* context, whereas the Definition 13 gives the context where the event cannot be added, and it is called *negative* context. These two sets are used to obtain the CD-relation.

Theorem 2. *Let $\mathcal{E} = \langle \mathsf{E}, \mathsf{S}, \mapsto, s_0 \rangle$ be an ea. Then $\mathcal{F}_{\mathsf{ea}}(\mathcal{E}) = (\mathsf{E}, \#, \gg)$ is a* CDES*, where $\#$ is the relation $\#_{\mathsf{ea}}$ of Definition 11, and for each $\mathsf{e} \in \mathsf{E}$ we have $\{(X, \emptyset) \mid X \in \mathcal{C}(\mathcal{E}, \mathsf{e})\} \cup \{(X, \{\mathsf{e}\}) \mid X \in \mathcal{I}(\mathcal{E}, \mathsf{e})\} \gg \mathsf{e}$. Furthermore $\mathcal{E} \equiv \mathcal{G}_{\text{CDES}}(\mathcal{F}_{\mathsf{ea}}(\mathcal{E}))$.*

The theorem has a main consequence, namely that event automata and CDES are equally expressive.

Example 7. Consider the RCES of the Example 6. The associated event automaton is the one depicted in the Example 1. It has no conflict as all the three events are present in a configuration together. The associated CD-relation, obtained using Definition 12 and Definition 13, is the following one, which is a little different from the one devised in the Example 1 as here it is obtained from an event automaton. $\{(\emptyset,\emptyset),(\{\mathsf{c}\},\emptyset),(\{\mathsf{c},\mathsf{b}\},\emptyset),(\{\mathsf{b}\},\{\mathsf{a}\})\} \gg \mathsf{a}$ because the set $\mathcal{C}(\mathsf{Conf}_{\text{RCES}}(R),\mathsf{a})$ contains the sets $\emptyset$, $\{\mathsf{c}\}$ and $\{\mathsf{c},\mathsf{b}\}$, whereas the set of the *negative context* $\mathcal{I}(\mathsf{Conf}_{\text{RCES}}(R),\mathsf{a})$ contains just $\{\mathsf{b}\}$, the one $\{(\emptyset,\emptyset),(\{\mathsf{c}\},\emptyset),(\{\mathsf{a},\mathsf{b}\},\emptyset),(\{\mathsf{a}\},\{\mathsf{b}\})\} \gg \mathsf{b}$ as $\mathcal{C}(\mathsf{Conf}_{\text{RCES}}(R),\mathsf{b})$ contains the sets $\emptyset$, $\{\mathsf{c}\}$ and $\{\mathsf{a},\mathsf{c}\}$, $\mathcal{I}(\mathsf{Conf}_{\text{RCES}}(R),\mathsf{b})$ contains $\{\mathsf{a}\}$, and finally $\{(\emptyset,\emptyset),(\{\mathsf{a}\},\emptyset),(\{\mathsf{b}\},\emptyset)\} \gg \mathsf{c}$ as $\mathcal{C}(\mathsf{Conf}_{\text{RCES}}(R),\mathsf{c})$ contains the sets $\emptyset$, $\{\mathsf{a}\}$ and $\{\mathsf{b}\}$, and $\mathcal{I}(\mathsf{Conf}_{\text{RCES}}(R),\mathsf{c})$ is the empty set.

As a consequence of the Theorem 2 we have the following result.

Corollary 1. *Let X be an event structure of type μ and let $\mathcal{G}_\mu(X)$ be the associated* ea*. Then $\mathcal{F}_{\mathsf{ea}}(\mathcal{G}_\mu(X))$ is* CDES*, and $X \equiv \mathcal{F}_{\mathsf{ea}}(\mathcal{G}_\mu(X))$.*

The construction identifies properly the *context* in which an event is allowed to happen, and this context becomes the main ingredient of the CD-relation, as the construction does not give the *causes* but just the context. If on the one hand this suggests that the context, rather than the causal dependencies, is the relevant ingredient, on the other hand it is less informative with respect to the usual causality definitions.

We review some kind of event structures, showing that a more informative CD-relation can be indeed obtained. We will focus only on few of them.

PES*:* In this case the idea is that causes of an event are just the set of events that should be present in the configuration.

Proposition 1. *Let $P = (\mathsf{E},\leq,\#)$ be a* PES*. Then $\mathcal{F}_{\text{PES}}(P) = (\mathsf{E},\#,\gg)$ is a* CDES*, where $\{(\emptyset,\lfloor \mathsf{e}\rfloor \setminus \{\mathsf{e}\})\} \gg \mathsf{e}$ for each $\mathsf{e} \in \mathsf{E}$. Furthermore $P \equiv \mathcal{F}_{\text{PES}}(P)$.*

This is not the unique way to associate to the causality relation $\leq$ of a PES the $\gg$ relation: one alternative would have been to add $\{(\emptyset,\{\mathsf{e}'\})\} \gg \mathsf{e}$ for each $\mathsf{e}' < \mathsf{e}$ and another one would be $\{(\lfloor \mathsf{e}\rfloor \setminus \{\mathsf{e}\},\emptyset)\} \gg \mathsf{e}$ showing that the events causally before e are indeed the context allowing the event e to happen.

Example 8. Consider the PES $(\{\mathsf{a},\mathsf{b},\mathsf{c}\},\leq,\#)$ where $\mathsf{a} \leq \mathsf{b}$ (we omit the reflexive part of the $\leq$ relation), $\mathsf{a} \# \mathsf{c}$ and $\mathsf{b} \# \mathsf{c}$. The event traces are ϵ, a, ab and c, and the associated configurations are $\emptyset$, $\{\mathsf{a}\}$, $\{\mathsf{a},\mathsf{b}\}$ and $\{\mathsf{c}\}$ (the $\mapsto_{\text{PES}}$ relation is obvious). The conflict relation is the same and the CD-relation is $\{(\emptyset,\emptyset)\} \gg \mathsf{a}$, $\{(\emptyset,\emptyset)\} \gg \mathsf{c}$ and $\{(\emptyset,\{\mathsf{a}\})\} \gg \mathsf{b}$. As noticed before we could have stipulated also $\{(\{\mathsf{a}\},\emptyset)\} \gg \mathsf{b}$ instead of $\{(\emptyset,\{\mathsf{a}\})\} \gg \mathsf{b}$ obtaining the same set of configurations and the same transition graph.

DCES*:* The intuition in this case consists in mixing the two approaches above.

Proposition 2. *Let* $D = (\mathsf{E}, \#, \rightarrow, \lhd, \blacktriangleright)$ *be a* DCES*.* $\mathcal{F}_{\text{DCES}}(D) = (\mathsf{E}, \#, \gg)$ *is a* CDES *where the relation* $\gg$ *is defined as* $\{(X, (\mathsf{ic}(\mathsf{e}) \setminus (\bigcup_{\mathsf{e}' \in X} \mathsf{Drop}(\mathsf{e}', \mathsf{e}))) \cup \bigcup_{\mathsf{e}' \in X} \mathsf{Add}(\mathsf{e}', \mathsf{e})) \mid X \subseteq \mathsf{GroMod}(\mathsf{e}) \cup \mathsf{ShrMod}(\mathsf{e}) \gg \mathsf{e}$ *for each* $\mathsf{e} \in \mathsf{E}$*. Furthermore* $D \equiv \mathcal{F}_{\text{DCES}}(G)$*.*

Example 9. Concerning the DCES of the example 4, the conflict relation is the one of the DCES whereas the CD-relation is $\{(\emptyset, \emptyset)\} \gg \mathsf{a}$, $\{(\emptyset, \emptyset)\} \gg \mathsf{b}$, $\{(\emptyset, \emptyset)\} \gg \mathsf{e}$, $\{(\emptyset, \emptyset)\} \gg \mathsf{d}$ and for c we have $\{(\emptyset, \{\mathsf{b}\}), (\{\mathsf{a}\}, \emptyset), (\{\mathsf{d}\}, \{\mathsf{b}, \mathsf{e}\}), (\{\mathsf{a}, \mathsf{d}\}, \{\mathsf{e}\})\} \gg \mathsf{c}$.

IES*:* In the case of IES there are two main observations: one, there is no conflict relation, and second, though there is some similarity between the $\vdash\!\!\circ$ relation and the $\gg$ relation, there is also a quite subtle difference. When adding an event e to a configuration of an IES, and we have $\vdash\!\!\circ(a, \mathsf{e}, A)$, one would simply add the pairs $(a, \{\mathsf{e}'\})$ for each $\mathsf{e}' \in A$ (as the events in A are pairwise conflicting) but this does not work in the case A is the empty set, as it has a different meaning in the $\vdash\!\!\circ$ relation with respect to the $\gg$ relation. In the former, it means that the event in a inhibits the event e, whereas in the latter the pair $(a, \emptyset)$ simply says that if the context a is present then there is no further event needed. Taking into account these differences, the translation is fairly simple. We first define the conflict relation and then the relation $\gg$, which is almost the same as the $\vdash\!\!\circ$ relation.

Proposition 3. *Let* $I = (\mathsf{E}, \vdash\!\!\circ)$ *be an* IES*.* $\mathcal{F}_{\text{IES}}(I) = (\mathsf{E}, \#, \gg)$ *is a* CDES*, where* $\mathsf{e} \# \mathsf{e}'$ *iff* $\vdash\!\!\circ(\{\mathsf{e}\}, \mathsf{e}', \emptyset)$ *and* $\vdash\!\!\circ(\{\mathsf{e}'\}, \mathsf{e}, \emptyset)$ *, and for each* $\mathsf{e} \in \mathsf{E}$*, if* $\vdash\!\!\circ(a, \mathsf{e}, A)$ *and* $A \neq \emptyset$ *then* $\{(\emptyset, \emptyset)\} \cup \{(a, \{\mathsf{e}'\}) \mid \mathsf{e}' \in A\} \gg \mathsf{e}$*, if* $\vdash\!\!\circ(a, \mathsf{e}, A)$ *and* $A = \emptyset$ *then* $\{(a, \{\mathsf{e}\})\} \gg \mathsf{e}$*. Furthermore* $I \equiv \mathcal{F}_{\text{IES}}(I)$*.*

Example 10. The IES of the example 5 induces the empty conflict relation, and the CD-relation is $\{(\emptyset, \emptyset)\} \gg \mathsf{a}$, $\{(\emptyset, \{\mathsf{a}\})\} \gg \mathsf{b}$ and $\{(\emptyset, \emptyset), (\{\mathsf{a}\}, \{\mathsf{b}\})\} \gg \mathsf{c}$.

Higher order causality: The comparison with event structures with higher-order dynamics of [14] is done indirectly, as these are equivalent to event structures with resolvable conflicts. In this approach the relations $\lhd$ and $\blacktriangleright$ are generalized to take into account set of modifiers, targets and contributions. The drawback is that the happening of an event implies a recalculation of these relation, similarly to what it is done in causal automata. In fact it is fairly obvious that given one simple step transition graph (meaning that a configuration is reached by another one adding just one event), it is always possible to obtain a CDES.

5 Conclusion

In this paper we have introduced a new brand of event structure where the main relation, the CD-relation, models the various conditions under which an event can be added to a subset of events. The relation is now defined as $\gg \subseteq \mathbf{2}^{\mathsf{A}} \times \mathsf{E}$, where $\mathsf{A} \subseteq \mathbf{2}^{\mathsf{E}} \times \mathbf{2}^{\mathsf{E}}$, thus it stipulates for each event which are the context-dependency pairs, but it can be easily generalized to subsets of events modeling precisely, when events happen together (as it is done in [23] or [26]). The focus is on the contexts in which an event can be added, which may change, rather that modeling the dependencies and how these may change. Here the choice is whether it is better to focus on dependencies (and how they may change) or on the context. The advantage of the latter is its generality, whereas the former may be useful in pointing out relations among events.

It should be clear that this kind of event structures is capable of modeling the same enabling situation for an event in various way, and it could be interesting to understand if there could be an informative way canonically. In fact, the canonical relation just focus on all the contexts in which an event can be added, and the dependency set is less informative. Thus finding a way to identify minimal contexts together with a set of dependencies may be useful, similarly to what it has been discussed when associating PES to CDES.

It remains to stress that CDES can be generalized not only allowing steps but also representing contexts in a richer way. Here we have considered contexts as subset of events, but they can have a richer structure. This would allow to characterize more precisely contexts, allowing, for instance, to drop the last requirement we have placed on DCES, as in this case the order in which the modifiers appear may influence the dependencies. Finally we observe that the idea of context is not new, for instance they have been considered in [17] or in [4], and a comparison with these should be considered.

In this paper we have considered various event structures, still some interesting notions remained out of the scope of this paper, like reversible event structures [19], but we are confident that our approach can be used also in the reversibility setting.

References

1. Arbach, Y., Karcher, D., Peters, K., Nestmann, U.: Dynamic causality in event structures. In: Graf, S., Viswanathan, M. (eds.) FORTE '15 Conference Proceedings. Lecture Notes in Computer Science, vol. 9039, pp. 83–97. Springer Verlag (2015)
2. Arbach, Y., Karcher, D., Peters, K., Nestmann, U.: Dynamic causality in event structures (technical report). CoRR **abs/1504.00512** (2015)
3. Arbach, Y., Karcher, D.S., Peters, K., Nestmann, U.: Dynamic Causality in Event Structures. Logical Methods in Computer Science **Volume 14, Issue 1** (Feb 2018)
4. Baldan, P., Bracciali, A., Bruni, R.: A semantic framework for open processes. Theoretical Computer Science **389**(3), 446–483 (2007)
5. Baldan, P., Busi, N., Corradini, A., Pinna, G.M.: Domain and event structure semantics for Petri nets with read and inhibitor arcs. Theoretical Computer Science **323**(1-3), 129–189 (2004)
6. Baldan, P., Corradini, A., Montanari, U.: Contextual Petri nets, asymmetric event structures and processes. Information and Computation **171**(1), 1–49 (2001)
7. Bartoletti, M., Cimoli, T., Pinna, G.M., Zunino, R.: Circular causality in event structures. Fundamenta Informaticae **134**(3-4), 219–259 (2014)

8. Boudol, G.: Flow Event Structures and Flow Nets. In: Semantics of System of Concurrent Processes. Lecture Notes in Computer Science, vol. 469, pp. 62–95. Springer Verlag (1990)
9. Gaifman, H.: Modeling concurrency by partial orders and nonlinear transition systems. In: de Bakker, J.W., de Roever, W.P., Rozenberg, G. (eds.) Linear Time, Branching Time and Partial Order in Logics and Models for Concurrency, REX Workshop. Lecture Notes in Computer Science, vol. 354, pp. 467–488. Springer Verlag (1989)
10. Gaifman, H., Pratt, V.R.: Partial order models of concurrency and the computation of functions. In: Gries, D. (ed.) LICS'87 Conference Proceedings. pp. 72–85. IEEE Computer Society (1987)
11. Gunawardena, J.: Geometric logic, causality and event structures. In: Baeten, J.C.M., Groote, J.F. (eds.) CONCUR'91 Conference Proceedings. Lecture Notes in Computer Science, vol. 527, pp. 266–280. Springer Verlag (1991)
12. Gunawardena, J.: Causal automata. Theoretical Computer Science **101**(2), 265–288 (1992)
13. Janicki, R., Koutny, M.: Semantics of inhibitor nets. Information and Computation **123**, 1–16 (1995)
14. Karcher, D.S., Nestmann, U.: Higher-order dynamics in event structures. In: Leucker, M., Rueda, C., Valencia, F.D. (eds.) ICTAC 2015 Conference Proceedings. Lecture Notes in Computer Science, vol. 9399, pp. 258–271. Springer Verlag (2015)
15. Langerak, R.: Bundle Event Structures: A Non-Interleaving Semantics for Lotos. In: Diaz, M., Groz, R. (eds.) FORTE '92 Conference Proceedings. IFIP Transactions, vol. C-10, pp. 331–346. North-Holland (1992)
16. Langerak, R., Brinksma, E., Katoen, J.: Causal ambiguity and partial orders in event structures. In: Mazurkiewicz, A.W., Winkowski, J. (eds.) CONCUR'97 Conference Proceedings. Lecture Notes in Computer Science, vol. 1243, pp. 317–331. Springer Verlag (1997)
17. Leifer, J.J., Milner, R.: Transition systems, link graphs and petri nets. Mathematical Structures in Computer Science **16**(6), 989–1047 (2006)
18. Nielsen, M., Plotkin, G., Winskel, G.: Petri Nets, Event Structures and Domains, Part 1. Theoretical Computer Science **13**, 85–108 (1981)
19. Phillips, I., Ulidowski, I.: Reversibility and asymmetric conflict in event structures. Journal of Logic and Algebraic Methods in Programming **84**(6), 781–805 (2015)
20. Pinna, G.M.: Event structures with disabling/enabling relation and event automata. Fundamenta Informaticae **73**(3), 409–430 (2006)
21. Pinna, G.M., Poigné, A.: On the nature of events. In: Havel, I.M., Koubek, V. (eds.) MFCS'92 Conference Proceedings. Lecture Notes in Computer Science, vol. 629, pp. 430–441. Springer Verlag (1992)
22. Pinna, G.M., Poigné, A.: On the nature of events: another perspective in concurrency. Theoretical Computer Science **138**(2), 425–454 (1995)
23. Pinna, G.M., Saba, A.: Modeling dependencies and simultaneity in membrane system computations. Theoretical Computer Science **431**, 13–39 (2012)
24. Reisig, W.: Petri Nets: An Introduction. EACTS Monographs on Theoretical Computer Science, Springer Verlag (1985)
25. Rensink, A.: Posets for configurations! In: Cleaveland, R. (ed.) CONCUR'92 Conference Proceedings. Lecture Notes in Computer Science, vol. 630, pp. 269–285. Springer Verlag (1992). https://doi.org/10.1007/BFb0084777

26. van Glabbeek, R.J., Plotkin, G.D.: Event structures for resolvable conflict. In: MFCS'04 Conference Proceedings. Lecture Notes in Computer Science, vol. 3153, pp. 550–561. Springer (2004)
27. Winskel, G.: Event Structures. In: Brauer, W., Reisig, W., Rozenberg, G. (eds.) Petri Nets: Central Models and Their Properties. Lecture Notes in Computer Science, vol. 255, pp. 325–392. Springer (1987)