Characterising spectra of equivalences for event structures, logically

Paolo Baldan[a], Daniele Gorla[b,*], Tommaso Padoan[a], Ivano Salvo[b]

[a] *University of Padua, Department of Mathematics, Italy*
[b] *Sapienza University of Rome, Department of Computer Science, Italy*

ABSTRACT

We present a logical characterisation of the equivalences in the spectrum for labelled Prime Event Structures (PESs) and use it for also studying how such spectrum changes when restricting to subclasses of event structures. We first show that a minor modification of the logic characterising hereditary history preserving bisimilarity induces PES isomorphism as logical equivalence. Then, we distill fragments of the logic that characterise all the equivalences in the aforementioned spectrum. In particular, we single out logics for interleaving/step/pomset trace equivalences and weak (pomset) history preserving bisimilarity, which were missing. Finally, we apply our logical characterisation to investigate how the spectrum simplifies when we restrict to subclasses of event structures: Coherence Spaces, where causality is absent, and Elementary Event Structures, where instead conflicts are not allowed. Inclusions between behavioural equivalences are proved by providing encodings between the corresponding sublogics, whereas the non-inclusion between equivalences is witnessed by using distinguishing formulae, i.e., by providing structures which are identified by an equivalence and distinguished by a formula in the logics of the other equivalence.

0. Introduction

Event structures [42,57] are one of the best known models for the formal treatment of true concurrency. Basically, they are collections of events, some of which are in conflict (i.e., the execution of an event forbids the execution of other events), while others are causally dependent (i.e., an event cannot be executed if it has not been preceded by other ones). Prime Event Structures (written PESs) are the earliest and simplest form of event structures, where causality is a partial order and all the conflicts of one event are inherited by all its causal successors. Events are typically labelled, to represent different occurrences of the same action.

Differently from what happens in the so-called interleaving approach, where the concurrent execution of events is identified with the non-deterministic choice between their sequentialisations, in the true concurrent approach concurrency is represented primitively. This can be particularly convenient or essential when one is interested in describing the causal history of computational steps, the flow of information or the level of parallelism; see, e.g., [56] for a reasoned survey on the use of such causal models. In the literature, event structures have been used to study concurrency in weak memory models

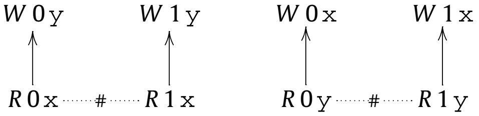
[*] Corresponding author.
E-mail addresses: baldan@math.unipd.it (P. Baldan), gorla@di.uniroma1.it (D. Gorla), padoan@math.unipd.it (T. Padoan), salvo@di.uniroma1.it (I. Salvo).

[13,36,47], for process mining and differencing [19], and to study atomicity [7,20], information flow [1] and trust [37,38,41] properties.

In order to grasp some intuition, let us hint at how PESs can be used to provide the semantics of the memory model of concurrent programming languages. Consider a simple programming language where all values are booleans, registers (ranged over by $\mathtt{r}$) are thread-local and variables (ranged over by $\mathtt{x}$ and $\mathtt{y}$) are global. In order to define the semantics compositionally, variable read can be defined as a choice among the possible values that might be read. Hence, the event structure for the two-threaded program

$$(\mathtt{r1} = \mathtt{x};\ \mathtt{y} = \mathtt{r1};) \mid\mid (\mathtt{r2} = \mathtt{y};\ \mathtt{x} = \mathtt{r2};)$$

is:

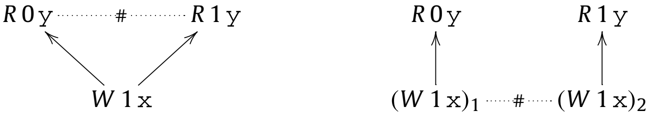


Here, events are represented by their labels: for instance, the label $R\,0\,\mathtt{x}$ denotes reading value 0 from variable $\mathtt{x}$.[1] Furthermore, arrows represent causality and # represent conflicts (only direct conflicts are represented explicitly, but if two events are in conflict, then all their causal consequences are so). In this setting, program executions correspond to *configurations*, i.e., conflict-free and causally closed sets of events (the presence of an event requires the presence of all its causes). A possible (maximal) configuration of the above PES is $\{R\,0\,\mathtt{x},\ W\,0\,\mathtt{y},\ R\,0\,\mathtt{y},\ W\,0\,\mathtt{x}\}$ arising, e.g., by performing the four events (corresponding to the given labels) exactly in this order. Despite being very basic, this representation allows one to capture interesting aspects of computations, like identifying the source (write operation) of a datum which is read. This can be helpful for tracing the flow of information or for ensuring the absence of cyclic dependencies (related to the presence of "thin-air reads" in memory models).

Also note that some subtleties intervene in the definition of such models. For example the following basic program

$$\mathtt{x} = \mathtt{1};\ \mathtt{r} = \mathtt{y}$$

could be modelled by one of the following two PESs



according to whether, in the execution of the program, one wants to observe the branching points (left-most PES, where the choice is done after writing variable $\mathtt{x}$) or just the possible sequences of actions (right-most PES, where the choice is moved to the beginning of the computation, thus leading to two conflicting occurrences of the same write action on $\mathtt{x}$, represented by events with the same label and different subscripts, the first followed by a read of value 0, the second by a read of value 1). This opens the questions of which model is "the" correct one or, at least, whether the two models are in some sense "equivalent" (we will see in Example 4 that the two models are equivalent only under notions of equivalence based on traces).

Indeed, in the setting of PESs, many equivalences have been developed, stemming from the notions of trace [35] and bisimulation [39]. In turn, these two families of equivalences depend on the unit of observation (a single event, a set of concurrent events – called *step* – or a set of events together with its causality and concurrency relations – called *pomset*), thus leading to six different equivalence relations: interleaving/step/pomset trace equivalences and interleaving/step/pomset bisimulation equivalences, respectively written $\approx_{\text{it}}$, $\approx_{\text{st}}$, $\approx_{\text{pt}}$, $\approx_{\text{ib}}$, $\approx_{\text{sb}}$, and $\approx_{\text{pb}}$ [6,49]. Moreover, stronger notions of bisimulation are obtained by imposing constraints on the causal structure of configurations related by a bisimulation: this leads to history preserving bisimilarity and its variants, namely weak, weak pomset and hereditary history preserving bisimilarity, respectively written $\approx_{\text{hb}}$, $\approx_{\text{whb}}$, $\approx_{\text{wphb}}$, and $\approx_{\text{hhb}}$ [5,17,18,50]. These equivalences together with the most discriminating one, i.e. PES isomorphism (written $\cong$), form a well known spectrum [21,51] (see Fig. 1 in Section 1).

On the logical side, various behavioural logics capable of expressing causal properties of computations have been proposed (see, e.g., [5,8,15,31–33,40,44,48], just to mention a few). In particular, a logical characterisation of some relevant behavioural equivalences in the true concurrent spectrum [51] has been provided using event-based logics [3,46] and interpreted over event structures; this means that formulae include variables which can be bound to events in computations.

In this paper we thoroughly investigate the true concurrent spectrum from the logical point of view, studying also the impact on such spectrum of confining the attention to subclasses of event structures, where either causality or conflict is

[1] Note that, in this model, register values are resolved via substitution and therefore do not appear.

removed. To this aim, we find it handier to work with a generalisation of the logic in [3], which predicates explicitly about dependencies between events.

As a first step, we need to complete the logical characterisation of the true concurrent spectrum, providing a uniform logical characterisation for all the equivalences mentioned above. In particular, all logics can be obtained as fragments of the logic characterising the most discriminating equivalence, i.e., PES isomorphism. Interestingly enough, by removing a well-formation constraint on formulae present in the logic for hereditary history preserving bisimilarity proposed in [3], we obtain a logic expressive enough to characterise PES isomorphism. A logic for isomorphism, apart from being useful to express very fine-grained system properties, is also interesting because it clearly singles out where the difference between isomorphism and hereditary-hp lies: they only differ when there are two (or more) conflicting identical branches. This testifies how close the two equivalences are: indeed, they remain distinct until conflict is present, whereas they collapse as soon as conflict is removed (see Section 4). Our characterizations extend the work in [3] where all bisimulation equivalences except for weak history preserving bisimilarities have been logically characterised. Then, for interleaving/step/pomset trace equivalences, the logic can be derived from the logic for the corresponding bisimilarity by simply removing negations and conjunctions (as it happens in process calculi).

By contrast, singling out fragments of the logic that capture weak history preserving bisimilarity and weak pomset history preserving bisimilarity is far from trivial. Intuitively, the distinguishing power of weak history preserving bisimilarity relies on the execution of events (or pomsets) with a specific label and on the check that the reached configurations are isomorphic as posets. As a consequence, the logic must contain formulae that offer exactly such distinguishing power. This requires some ingenuity, since the naming of variables and the order in which they have been bound must be irrelevant, while their labels and causal relations matter. A similar intuition is at the heart of the logic provided in [45] for weak history preserving bisimilarity, but our formalisation is necessarily quite different as their logic uses backward modalities, absent in our logic.

In the second part of the paper, we apply our logical characterisation to investigate the impact on the spectrum by considering two subclasses of PESs, along the lines of [26,27]. The first one is obtained by removing the causality relation and this essentially leads to (the web of) *coherence spaces* [25] (written CSs). CSs together with stable functions play a key role in building stable domains in the semantics of linear logic [25] and typed lambda-calculus [9,10]. The second one is obtained by removing the conflict relation; this leads to *elementary event structures* [42] (written EESs), that naturally arise as the deterministic subclass of PESs (see, e.g., [42,43,51]). As shown in [26,27], in both cases the spectrum turns out to be simplified, since some notions of equivalence coincide in the simplified settings. For CSs, the spectrum is simplified to a chain as depicted in Fig. 11 in Subsection 4.1: all trace equivalences coincide and represent the coarsest notion; they properly include bisimilarities that all coincide, except for hereditary history preserving bisimilarity. The latter, in turn is properly refined by PES isomorphism. For EESs pomset trace, pomset bisimilarity and all history preserving bisimilarities coincide with isomorphism, whereas interleaving/step trace/bisimulation equivalences are as in the general spectrum, as depicted in Fig. 12 of Subsection 4.2.

These results were known from [26,27], where they were proved for so called *finitary* PESs (i.e., PESs that admit a finite number of configurations of every finite cardinality). Here we provide a logical account of the same results. Inclusions between behavioural equivalences are shown by providing encodings between the corresponding sublogics, whereas the non-inclusion between equivalences is witnessed by using distinguishing formulae, i.e., by providing structures which are identified by an equivalence and distinguished by a formula in the logics of the other equivalence. Since the logical characterisation works, as usual, for *image-finite* PESs (i.e. structures where every configuration can reach only finitely many others by means of a transition with a fixed label), our results hold in this setting, that slightly generalises that of finitary PESs. Notice that in [26,27] the case of countable EESs without finitariness assumption was also discussed. Since logics usually assume to have some form of finitariness, we do not consider general EESs here.

This paper is a completion of both [3] and [26,27]. With respect to [3], here we provide a uniform treatment of all the logics for the 11 equivalences of Fig. 1, including all trace equivalences, isomorphism and weak (pomset) history preserving bisimilarity: all these results are new in the framework of [3]. Furthermore, as mentioned above, we slightly strengthen the characterisations already in [3] by proving all our results in the framework of image-finite structures (instead of the smaller class of finitely-branching structures).

The rest of the paper is organised as follows. In Section 1, we recall the definition of the behavioural equivalences in the spectrum for PESs, as reported in [21]. In Section 2, we introduce the logic and a few basic derived operators. In Section 3, we prove that the logic presented characterises isomorphism, and then distill from it all the sublogics needed to characterise the behavioural equivalences introduced in Section 1. In Section 4, we logically study the impact on the spectrum of removing either causality or conflict. Section 5 concludes the paper. To streamline reading, some technical material is put into the Appendix.

1. Background: prime event structures and their equivalence spectrum

We start by briefly recalling some notions and results from the theory of event structures [42], by following the presentation in [51].

Definition 1 *(Prime Event Structures [42,57]).* A (labelled) *prime event structure* (PES, for short) over an alphabet $\mathcal{A}$ is a tuple $\mathcal{E} = (E, \leq, \#, l)$ such that:

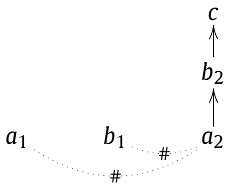
- E is a set of *events*;
- $\leq \subseteq E \times E$ is the *causality* relation, i.e. a partial order such that, for all $e \in E$, the set $\lfloor e \rfloor = \{e' : e' \leq e\}$ is finite;
- $\# \subseteq E \times E$ is the *conflict* relation, i.e. an irreflexive and symmetric relation such that, for all $e, e', e'' \in E$, if $e \leq e'$ and $e \# e''$, then $e' \# e''$;
- $l : E \to \mathcal{A}$ is the *labeling* function.

The PESs $\mathcal{E} = (E, \leq_E, \#_E, l_E)$ and $\mathcal{F} = (F, \leq_F, \#_F, l_F)$ are *isomorphic*, written $\mathcal{E} \cong \mathcal{F}$, if there exists bijection $f : E \to F$ that respects relations and labelling (i.e., for all $e, e' \in E$, it holds $e \leq_E e'$ iff $f(e) \leq_F f(e')$, $e \#_E e'$ iff $f(e) \#_F f(e')$ and $l_E(e) = l_F(f(e))$).

Intuitively, $e' \leq e$ means that e cannot happen before e' (so, the execution of e causally depends on the execution of e'), whereas $e \# e'$ means that e and e' are mutually exclusive (so, the execution of one prevents the execution of the other). The requirement for $\lfloor e \rfloor$ to be finite ensures that every event can be executed in a finite amount of time (i.e., after the execution of finitely many events).

Definition 2 *(Consistency, Concurrency).* Let $\mathcal{E}$ be a PES. We say that $e, e' \in E$ are *consistent*, written $e \frown e'$, if $\neg(e \# e')$. A subset $X \subseteq E$ is called *consistent* if $e \frown e'$ for all $e, e' \in X$. We say that e and e' are *concurrent*, written $e \parallel e'$, if $\neg(e \leq e')$, $\neg(e' \leq e)$ and $\neg(e \# e')$.

Following the notation used in the introduction, we shall represent events in a PES by their labels, possibly numbered in order to distinguish events with the same label (numbering proceeds from left to right and from bottom to top); causality is represented by upwards arrows and conflicts by #-labelled dotted edges. Moreover, for the sake of simplicity, since causality is transitive, we shall usually represent the causality relation by its transitive reduction, i.e., only direct causes are depicted. For instance, the following picture



denotes a PES $\mathcal{E} = (E, \leq, \#, l)$ where $E = \{a_1, a_2, b_1, b_2, c\}$, causality is given by $e \leq e$ ($\forall e \in E$), $a_2 \leq b_2$, $a_2 \leq c$, $b_2 \leq c$, # is the symmetric closure of $\{(a_1, a_2), (b_1, a_2), (a_1, b_2), (b_1, b_2), (a_1, c), (b_1, c)\}$, $l(a_1) = l(a_2) = a$, $l(b_1) = l(b_2) = b$ and $l(c) = c$.

The possible states that the system modelled by a PES can pass through during its evolution are defined as follows.

Definition 3 *(Configurations).* Let $\mathcal{E} = (E, \leq, \#, l)$ be a PES. A *configuration* of $\mathcal{E}$ is a finite set $C \subseteq E$ such that

- $\neg(e \# e')$, for every $e, e' \in C$; and
- $\lfloor e \rfloor \subseteq X$, for every $e \in C$.

We denote with $\mathit{Conf}(\mathcal{E})$ the set of configurations of $\mathcal{E}$.

The way in which (the system modelled by) a PES evolves is usually given through some *labelled transition system* (LTS) over configurations. The idea is that inclusion between configurations represents computational extension, i.e., when C, C' are configurations and $C \subseteq C'$, then C can evolve to C', by performing the events in $C' \setminus C$. Depending on the number and structure of the extension that can be realised in one single transition, we obtain different transition systems and, correspondingly, different notions of behavioural equivalences, as detailed below.

In the simplest case, a transition consists of the execution of one single event. Given configurations $C, C' \in \mathit{Conf}(\mathcal{E})$, we write $C \xrightarrow{a} C'$ whenever $C \subseteq C'$ and $C' \setminus C = \{e\}$, with $l(e) = a$. In this case, we say that the event e is *enabled* in C. Notation $C \rightarrow$ (resp., $C \not\rightarrow$) means that there exist a and C' (resp., no a and C') such that $C \xrightarrow{a} C'$.

The two most basic equivalences we consider are derived from process algebras and are *interleaving trace* and *bisimulation* equivalence.

Definition 4 *(Interleaving trace equivalence [35]).* Let $\mathcal{E}$ be a PES. A *(sequential) trace* of $\mathcal{E}$ is a sequence $a_1 \ldots a_k \in \mathcal{A}^*$ such that there exist $C_0, \ldots, C_k \in \mathit{Conf}(\mathcal{E})$ with $C_0 = \varnothing$ and $C_i \xrightarrow{a_{i+1}} C_{i+1}$, for $i = 0, \ldots, k-1$; by convention, the empty trace ϵ arises when $k = 0$. We denote with $\mathit{SeqTr}(\mathcal{E})$ the set of all sequential traces of $\mathcal{E}$.

The PESs $\mathcal{E}$ and $\mathcal{F}$ are *interleaving trace equivalent*, written $\mathcal{E} \approx_{\text{it}} \mathcal{F}$, if $\mathit{SeqTr}(\mathcal{E}) = \mathit{SeqTr}(\mathcal{F})$.

Definition 5 *(Interleaving bisimilarity [39]).* Let $\mathcal{E}$ and $\mathcal{F}$ be PESs. A relation $R \subseteq \mathit{Conf}(\mathcal{E}) \times \mathit{Conf}(\mathcal{F})$ is an *interleaving bisimulation* between $\mathcal{E}$ and $\mathcal{F}$ if:

- $(\emptyset, \emptyset) \in R$;
- if $(C, D) \in R$ and $C \xrightarrow{a} C'$, then $D \xrightarrow{a} D'$, for some D' such that $(C', D') \in R$;
- if $(C, D) \in R$ and $D \xrightarrow{a} D'$, then $C \xrightarrow{a} C'$, for some C' such that $(C', D') \in R$.

When there is such R, we say that $\mathcal{E}$ and $\mathcal{F}$ are *interleaving bisimilar*, and write $\mathcal{E} \approx_{\text{ib}} \mathcal{F}$.

Transitions involving one single action can be generalized to *steps*, i.e., sets of events that can be executed concurrently. A step transition will be labelled with the (finite) multiset of labels corresponding to the executed events. We write $C \xrightarrow{\{\!|a_1 \dots a_n|\!\}} C'$ if $C \subseteq C'$, $C' \setminus C = \{e_1, \dots, e_n\}$, $\forall i \neq j \,.\, e_i \parallel e_j$, and $\{\!|a_1 \dots a_n|\!\}$ denotes the multiset over $\mathcal{A}$ formed by the labels a_i of the events e_i. This yields the obvious generalization of interleaving trace and bisimulation equivalence, where the *step traces* of a PES $\mathcal{E}$, denoted $\mathit{StepTr}(\mathcal{E})$, are defined as expected (i.e., like sequential traces, but with steps in place of single events).

Definition 6 *(Step trace equivalence [49]).* The PESs $\mathcal{E}$ and $\mathcal{F}$ are *step trace equivalent*, written $\mathcal{E} \approx_{\text{st}} \mathcal{F}$, if $\mathit{StepTr}(\mathcal{E}) = \mathit{StepTr}(\mathcal{F})$.

Definition 7 *(Step bisimilarity [49]).* Let $\mathcal{E}$ and $\mathcal{F}$ be PESs. A relation $R \subseteq \mathit{Conf}(\mathcal{E}) \times \mathit{Conf}(\mathcal{F})$ is a *step bisimulation* between $\mathcal{E}$ and $\mathcal{F}$ if:

- $(\emptyset, \emptyset) \in R$;
- if $(C, D) \in R$ and $C \xrightarrow{A} C'$, then $D \xrightarrow{A} D'$, for some D' such that $(C', D') \in R$;
- if $(C, D) \in R$ and $D \xrightarrow{A} D'$, then $C \xrightarrow{A} C'$, for some C' such that $(C', D') \in R$.

When there is such R, we say that $\mathcal{E}$ and $\mathcal{F}$ are *step bisimilar*, and write $\mathcal{E} \approx_{\text{sb}} \mathcal{F}$.

An equivalence finer than step semantics can be obtained by allowing a single transition to perform a generic consistent set of events, possibly not concurrent. Given configurations $C, C' \in \mathit{Conf}(\mathcal{E})$ with $C \subsetneq C'$, observe that $Y = C' \setminus C$ can be seen as a partially ordered set (*poset*, for short), with the ordering given by $\leq$. We write $\mathit{poset}(Y)$ to denote the labelled poset $(Y, \leq\!|_Y, l|_Y)$, where $\leq\!|_Y$ and $l|_Y$ are the restrictions of $\leq$ and l to Y. A more abstract view is obtained by replacing events with their labels. This turns a poset into a so-called partially ordered multiset (*pomset*, for short). Formally, the pomset associated to Y, written $\mathit{pomset}(Y)$, is the isomorphism class of $\mathit{poset}(Y)$. We write $C \xrightarrow{p} C'$ when $p = \mathit{pomset}(C' \setminus C)$.

Clearly, also entire configurations can be seen as posets and thus, given a configuration C, we can consider $\mathit{pomset}(C)$, the pomset corresponding to C.

Definition 8 *(Pomset trace equivalence [6]).* Let $\mathcal{E}$ be a PES. The *pomset trace language* of $\mathcal{E}$ is $\mathit{Pom}(\mathcal{E}) = \{\mathit{pomset}(C) \mid C \in \mathit{Conf}(\mathcal{E})\}$.

The PESs $\mathcal{E}$ and $\mathcal{F}$ are *pomset trace equivalent*, written $\mathcal{E} \approx_{\text{pt}} \mathcal{F}$, if $\mathit{Pom}(\mathcal{E}) = \mathit{Pom}(\mathcal{F})$.

Notice that a pomset trace, differently from interleaving and step traces, consists of a single pomset, not of a sequence of pomsets. This is how pomset traces have been defined in [21], since it simplifies the treatment and it leads to the same equivalence (intuitively because a sequence of pomsets can be always captured by a single one and because trace semantics forgets about the branching points which might occur between two transitions).

Definition 9 *(Pomset bisimilarity [6]).* Let $\mathcal{E}$ and $\mathcal{F}$ be PESs. A relation $R \subseteq \mathit{Conf}(\mathcal{E}) \times \mathit{Conf}(\mathcal{F})$ is a *pomset bisimulation* between $\mathcal{E}$ and $\mathcal{F}$ if:

- $(\emptyset, \emptyset) \in R$;
- if $(C, D) \in R$ and $C \xrightarrow{p} C'$, then $D \xrightarrow{p} D'$, for some D' such that $(C', D') \in R$;
- if $(C, D) \in R$ and $D \xrightarrow{p} D'$, then $C \xrightarrow{p} C'$, for some C' such that $(C', D') \in R$.

When there is such R, we say that $\mathcal{E}$ and $\mathcal{F}$ are *pomset bisimilar*, and write $\mathcal{E} \approx_{\text{pb}} \mathcal{F}$.

An orthogonal way to generalise interleaving bisimilarity is to keep track of the causal dependencies inside the configurations, and only relate configurations with the same causal history. This is done by requiring that the two configurations have isomorphic associated posets, i.e., that they represent the same pomset.

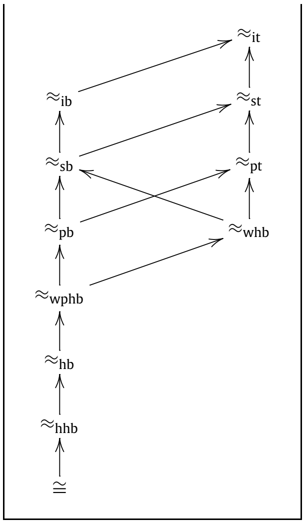


Fig. 1. The spectrum of equivalences for PESs, where '$\rightarrow$' means '$\subset$'.

Definition 10 *(Weak history preserving bisimilarities [17,51]).* Let $\mathcal{E}$ and $\mathcal{F}$ be PESs. A relation $R \subseteq Conf(\mathcal{E}) \times Conf(\mathcal{F})$ is a *weak history preserving bisimulation* between $\mathcal{E}$ and $\mathcal{F}$ if

- $(\varnothing, \varnothing) \in R$, and
- if $(C, D) \in R$ then
  - $pomset(C) = pomset(D)$;
  - if $C \xrightarrow{a} C'$, then $D \xrightarrow{a} D'$, for some D' such that $(C', D') \in R$;
  - if $D \xrightarrow{a} D'$, then $C \xrightarrow{a} C'$, for some C' such that $(C', D') \in R$.

When there is such R, we say that $\mathcal{E}$ and $\mathcal{F}$ are *weak history preserving bisimilar* (*weak h-bisimilar*) and write $\mathcal{E} \approx_{\mathrm{whb}} \mathcal{F}$. The definition of *weak pomset h-bisimilarity*, denoted $\approx_{\mathrm{wphb}}$, is obtained from the above by using pomset transitions.

Observe that one could also define a notion of weak step h-bisimilarity. However, it is not difficult to see that weak step h-bisimilarity would coincide with weak h-bisimilarity, since, as shown in [21], a weak h-bisimulation is also a step bisimulation.

A finer equivalence is obtained by requiring in the definition above that the isomorphism relating the target configurations C' and D' extends the isomorphism relating the source configurations C and D. This leads to the following definition.

Definition 11 *(History preserving bisimilarity [18,50]).* Let $\mathcal{E}$ and $\mathcal{F}$ be PESs. A relation $R \subseteq Conf(\mathcal{E}) \times Conf(\mathcal{F}) \times 2^{Conf(\mathcal{E}) \times Conf(\mathcal{F})}$ is a *history preserving bisimulation* between $\mathcal{E}$ and $\mathcal{F}$ if

- $(\varnothing, \varnothing, \varnothing) \in R$, and
- if $(C, D, f) \in R$ then
  - $f : poset(C) \rightarrow poset(D)$ is an isomorphism;
  - if $C \xrightarrow{a} C'$, then $D \xrightarrow{a} D'$, for some D' such that $(C', D', f') \in R$, where $f'|_C = f$;
  - if $D \xrightarrow{a} D'$, then $C \xrightarrow{a} C'$, for some C' such that $(C', D', f') \in R$, where $f'|_C = f$.

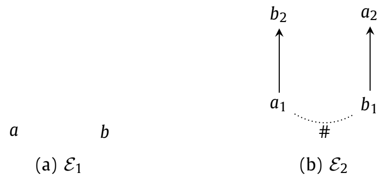
When such a relation R exists, we say that $\mathcal{E}$ and $\mathcal{F}$ are *history preserving (h-)bisimilar* and write $\mathcal{E} \approx_{\mathrm{hb}} \mathcal{F}$.

The notion of h-bisimulation can be made even finer by also asking for a 'backwards' bisimulation game, along the lines of back-and-forth bisimulation [16].

Definition 12 *(Hereditary h-bisimilarity [5]).* Let $\mathcal{E}$ and $\mathcal{F}$ be PESs. A h-bisimulation R between $\mathcal{E}$ and $\mathcal{F}$ is *hereditary* if, for every $(C, D, f) \in R$, it holds that $C' \xrightarrow{a} C$ implies $(C', f(C'), f|_{C'}) \in R$ and $D' \xrightarrow{a} D$ implies $(f^{-1}(D'), D', f|_{f^{-1}(D')}) \in R$.

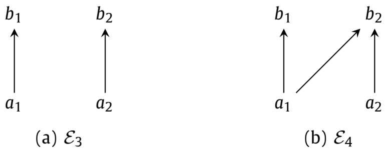
When such a relation R exists, we say that $\mathcal{E}$ and $\mathcal{F}$ are *hereditary history preserving bisimilar (hereditary h-bisimilar)* and write $\mathcal{E} \approx_{\mathrm{hhb}} \mathcal{F}$.

All the equivalences presented so far form a well known spectrum [21,51], depicted in Fig. 1 (the only inclusions that are not present in [51] are $\approx_{\mathrm{whb}} \subset \approx_{\mathrm{sb}}$ and $\approx_{\mathrm{whb}} \subset \approx_{\mathrm{pt}}$, that are proved in [21]). We next provide a series of examples (taken



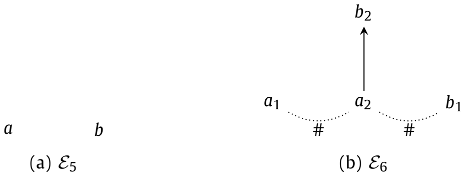
(a) $\mathcal{E}_1$ (b) $\mathcal{E}_2$

Fig. 2. $\mathcal{E}_1 \approx_{\mathrm{ib}} \mathcal{E}_2$ (hence $\mathcal{E}_1 \approx_{\mathrm{it}} \mathcal{E}_2$) but $\mathcal{E}_1 \not\approx_{\mathrm{st}} \mathcal{E}_2$ (hence $\mathcal{E}_1 \not\approx_{\mathrm{sb}} \mathcal{E}_2$).



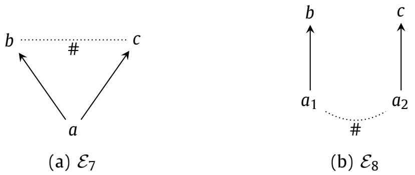
(a) $\mathcal{E}_3$ (b) $\mathcal{E}_4$

Fig. 3. $\mathcal{E}_3 \approx_{\mathrm{st}} \mathcal{E}_4$ (hence $\mathcal{E}_3 \approx_{\mathrm{it}} \mathcal{E}_4$) but $\mathcal{E}_3 \not\approx_{\mathrm{ib}} \mathcal{E}_4$ (hence $\mathcal{E}_3 \not\approx_{\mathrm{sb}} \mathcal{E}_4$).



(a) $\mathcal{E}_5$ (b) $\mathcal{E}_6$

Fig. 4. $\mathcal{E}_5 \approx_{\mathrm{sb}} \mathcal{E}_6$ (hence $\mathcal{E}_5 \approx_{\mathrm{st}} \mathcal{E}_6$) but $\mathcal{E}_5 \not\approx_{\mathrm{pt}} \mathcal{E}_6$ (hence $\mathcal{E}_5 \not\approx_{\mathrm{pb}} \mathcal{E}_6$).



(a) $\mathcal{E}_7$ (b) $\mathcal{E}_8$

Fig. 5. $\mathcal{E}_7 \approx_{\mathrm{pt}} \mathcal{E}_8$ (hence $\mathcal{E}_7 \approx_{\mathrm{it}} \mathcal{E}_8$ and $\mathcal{E}_7 \approx_{\mathrm{st}} \mathcal{E}_8$) but $\mathcal{E}_7 \not\approx_{\mathrm{ib}} \mathcal{E}_8$ (hence $\mathcal{E}_7 \not\approx_{\mathrm{sb}} \mathcal{E}_8$ and $\mathcal{E}_7 \not\approx_{\mathrm{pb}} \mathcal{E}_8$).

from the literature) that witness the strictness of the inclusions in Fig. 1. In Section 3.4, we shall provide a distinguishing logical formula for each of these examples; this will also show the usefulness of our logics as tools for reasoning on PESs and their equivalences.

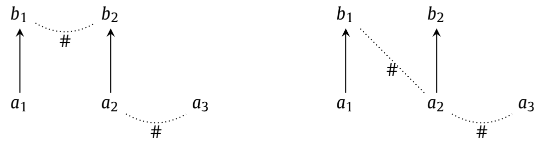
Example 1. Consider the two PESs in Fig. 2, taken from [12]. It can be easily checked that $\mathcal{E}_1 \approx_{\mathrm{ib}} \mathcal{E}_2$ (and hence $\mathcal{E}_1 \approx_{\mathrm{it}} \mathcal{E}_2$). By contrast, $\mathcal{E}_1 \not\approx_{\mathrm{st}} \mathcal{E}_2$ (and hence $\mathcal{E}_1 \not\approx_{\mathrm{sb}} \mathcal{E}_2$): indeed, $\mathcal{E}_1$ can perform the step $\{\!|a\ b|\!\}$ whereas $\mathcal{E}_2$ cannot.

Example 2. Consider the two PESs in Fig. 3 taken from [26, Prop. 8]. It can be easily checked that $\mathcal{E}_3 \approx_{\mathrm{st}} \mathcal{E}_4$ (and hence $\mathcal{E}_3 \approx_{\mathrm{it}} \mathcal{E}_4$). However, $\mathcal{E}_3 \not\approx_{\mathrm{ib}} \mathcal{E}_4$ (and hence $\mathcal{E}_3 \not\approx_{\mathrm{sb}} \mathcal{E}_4$): indeed, in $\mathcal{E}_3$ after every a-labelled event, we can always perform an event labelled b, whereas this is not the case for a_2 in $\mathcal{E}_4$.

Example 3. Consider the two PESs in Fig. 4, taken from [51, Ex. 7.2]. It can be easily checked that $\mathcal{E}_5 \approx_{\mathrm{sb}} \mathcal{E}_6$ (and hence $\mathcal{E}_5 \approx_{\mathrm{st}} \mathcal{E}_6$). By contrast, $\mathcal{E}_5 \not\approx_{\mathrm{pt}} \mathcal{E}_6$ (and hence $\mathcal{E}_5 \not\approx_{\mathrm{pb}} \mathcal{E}_6$): indeed, $\mathcal{E}_6$ can perform a pomset transition labelled with $\overset{b}{\underset{a}{\uparrow}}$ whereas $\mathcal{E}_5$ cannot.

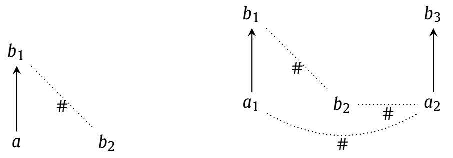
Example 4. Consider the two PESs in Fig. 5, taken from [51, Sect. 8]. It can be easily checked that $\mathcal{E}_7 \approx_{\mathrm{pt}} \mathcal{E}_8$ (and hence $\mathcal{E}_7 \approx_{\mathrm{it}} \mathcal{E}_8$ and $\mathcal{E}_7 \approx_{\mathrm{st}} \mathcal{E}_8$) but $\mathcal{E}_7 \not\approx_{\mathrm{ib}} \mathcal{E}_8$ (and hence $\mathcal{E}_7 \not\approx_{\mathrm{sb}} \mathcal{E}_8$ and $\mathcal{E}_7 \not\approx_{\mathrm{pb}} \mathcal{E}_8$). Indeed, this is the well-known Milner's example $a(b+c)$ versus $ab+ac$, used for distinguishing bisimulations from trace equivalences: the two are not bisimilar because, after executing a, $\mathcal{E}_7$ can choose between b and c, whereas $\mathcal{E}_8$ cannot.

Example 5. Consider the two PESs in Fig. 6, taken from [51, Ex. 9.3]. It can be checked that $\mathcal{E}_9 \approx_{\mathrm{whb}} \mathcal{E}_{10}$ (and hence $\mathcal{E}_9 \approx_{\mathrm{pt}} \mathcal{E}_{10}$). By contrast, $\mathcal{E}_9 \not\approx_{\mathrm{pb}} \mathcal{E}_{10}$ (and hence $\mathcal{E}_9 \not\approx_{\mathrm{wphb}} \mathcal{E}_{10}$): if $\mathcal{E}_{10}$ executes a_2, it can no more perform a pomset transition labelled with $\overset{b}{\underset{a}{\uparrow}}$; by contrast, after every a possibly used by $\mathcal{E}_9$ for replying, such a pomset transition is always executable.



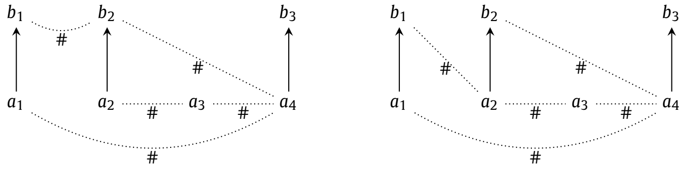
(a) $\mathcal{E}_9$ (b) $\mathcal{E}_{10}$

Fig. 6. $\mathcal{E}_9 \approx_{\text{whb}} \mathcal{E}_{10}$ (hence $\mathcal{E}_9 \approx_{\text{pt}} \mathcal{E}_{10}$) but $\mathcal{E}_9 \not\approx_{\text{pb}} \mathcal{E}_{10}$ (hence $\mathcal{E}_9 \not\approx_{\text{wphb}} \mathcal{E}_{10}$).



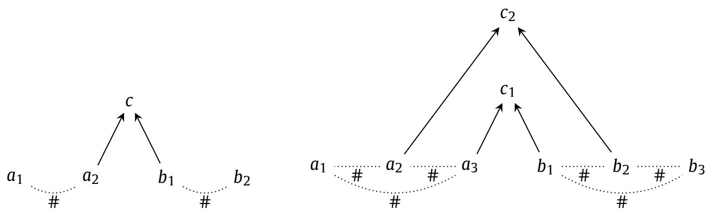
(a) $\mathcal{E}_{11}$ (b) $\mathcal{E}_{12}$

Fig. 7. $\mathcal{E}_{11} \approx_{\text{pb}} \mathcal{E}_{12}$ (hence $\mathcal{E}_{11} \approx_{\text{pt}} \mathcal{E}_{12}$ and $\mathcal{E}_9 \approx_{\text{sb}} \mathcal{E}_{10}$) but $\mathcal{E}_{11} \not\approx_{\text{whb}} \mathcal{E}_{12}$ (hence $\mathcal{E}_{11} \not\approx_{\text{wphb}} \mathcal{E}_{12}$).



(a) $\mathcal{E}_{13}$ (b) $\mathcal{E}_{14}$

Fig. 8. $\mathcal{E}_{13} \approx_{\text{wphb}} \mathcal{E}_{14}$ (hence $\mathcal{E}_{13} \approx_{\text{whb}} \mathcal{E}_{14}$) but $\mathcal{E}_{13} \not\approx_{\text{hb}} \mathcal{E}_{14}$.



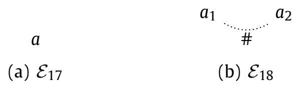
(a) $\mathcal{E}_{15}$ (b) $\mathcal{E}_{16}$

Fig. 9. $\mathcal{E}_{15} \approx_{\text{hb}} \mathcal{E}_{16}$ but $\mathcal{E}_{15} \not\approx_{\text{hhb}} \mathcal{E}_{16}$.

Example 6. Consider the two PESs in Fig. 7, taken from [54]. It can be checked that $\mathcal{E}_{11} \approx_{\text{pb}} \mathcal{E}_{12}$ (and hence $\mathcal{E}_{11} \approx_{\text{pt}} \mathcal{E}_{12}$ and $\mathcal{E}_9 \approx_{\text{sb}} \mathcal{E}_{10}$). Instead, $\mathcal{E}_{11} \not\approx_{\text{whb}} \mathcal{E}_{12}$ (and hence $\mathcal{E}_{11} \not\approx_{\text{wphb}} \mathcal{E}_{12}$): indeed, if $\mathcal{E}_{12}$ performs a_2, then $\mathcal{E}_{11}$ must reply with a; however, now in $\mathcal{E}_{11}$ we can perform b_1 that leads to the pomset $a\ b$, whereas in $\mathcal{E}_{12}$ the only possible reply is with b_3 that leads to the pomset $\overset{b}{\underset{a}{\uparrow}}$.

Example 7. Consider the two PESs in Fig. 8, taken from [51, Ex. 9.4]. It can be checked that $\mathcal{E}_{13} \approx_{\text{wphb}} \mathcal{E}_{14}$ (hence $\mathcal{E}_{13} \approx_{\text{whb}} \mathcal{E}_{14}$). However, $\mathcal{E}_{13} \not\approx_{\text{hb}} \mathcal{E}_{14}$. To see this, let us perform in $\mathcal{E}_{13}$ first a_1 and then a_2; now, if we perform b_1 we obtain the poset $a_2\ \overset{b_1}{\underset{a_1}{\uparrow}}$, whereas if we perform b_2 we obtain the poset $a_1\ \overset{b_2}{\underset{a_2}{\uparrow}}$. This situation cannot be found in $\mathcal{E}_{14}$: whatever the reply in $\mathcal{E}_{14}$ to the two initial a's is (i.e., a_1 and a_2 in any order, or a_1 and a_3 in any order, or a_2 and a_4 in any order), we cannot then execute both an event labelled b causally dependent from the first a and another causally dependent from the second a. By contrast, notice that, by changing the isomorphism (as the two weak h-bisimilarities allow to do), a proper reply to the challenge from $\mathcal{E}_{13}$ can be found in $\mathcal{E}_{14}$.

Example 8. Consider the two PESs in Fig. 9, taken from [22]. It can be checked that $\mathcal{E}_{15} \approx_{\text{hb}} \mathcal{E}_{16}$ but $\mathcal{E}_{15} \not\approx_{\text{hhb}} \mathcal{E}_{16}$. To see this, first consider the sequence of actions a_3 and b_1 in $\mathcal{E}_{16}$; these must be replied to by a_2 and b_1 in $\mathcal{E}_{15}$. Then, let us perform a backward step on both sides that removes the a's and then perform a_2 in $\mathcal{E}_{16}$; this must be replied to by a_1 in $\mathcal{E}_{15}$. Now, let us perform a backward step on both sides that removes the b's; after this, we are left with the pair of configurations $\{a_1\}$ in $\mathcal{E}_{15}$ and $\{a_2\}$ in $\mathcal{E}_{16}$. If we now perform b_2 in $\mathcal{E}_{16}$, we have no suitable reply in $\mathcal{E}_{15}$ because $\{a_2, b_2\}$ in $\mathcal{E}_{16}$ can perform c_2 whereas, whatever b we perform from $\{a_1\}$ in $\mathcal{E}_{15}$, we can never perform c.



(a) $\mathcal{E}_{17}$ (b) $\mathcal{E}_{18}$

Fig. 10. $\mathcal{E}_{17} \approx_{\text{hhb}} \mathcal{E}_{18}$ but $\mathcal{E}_{17} \ncong \mathcal{E}_{18}$.

Example 9. Consider the two PESs in Fig. 10, taken from [5]. It can be easily checked that $\mathcal{E}_{17} \approx_{\text{hhb}} \mathcal{E}_{18}$ but, trivially, $\mathcal{E}_{17} \ncong \mathcal{E}_{18}$.

To conclude, in the rest of the paper, we will assume all PESs to be countable (i.e., with a countable number of events). Moreover, we will focus on *image-finite* PESs, where for every configuration C and label $a \in \mathcal{A}$, there exist a finite number of configurations C' such that $C \xrightarrow{a} C'$ (observe that image-finiteness with respect to transitions consisting of single events implies image-finiteness also for step and pomset transitions). As already mentioned, the results in [26] were proved for *finitary* PESs, i.e., PESs that admit a finite number of configurations of every finite cardinality. Note that image-finite PESs properly include finitary PESs. For instance, assume that the set of labels is $\mathcal{A} = \{a_i \mid i \in \mathbb{N}\}$ and consider the PES $\mathcal{E}$ with events $E = \{e_i \mid i \in \mathbb{N}\}$, labelling $l(e_i) = a_i$, and empty conflict and causality. It is easy to see that $\mathcal{E}$ is image-finite, but not finitary. In this view, the results presented in Section 4 extend those of [26,27]. As usual, we could get rid of any finitariness assumption, but we would then need to include infinite conjunctions in our logics to characterise bisimilarities.

2. Logics for event structures

We present the logic that we will use to study the spectrum of true concurrent behavioural equivalences. It is a smooth variation of the logic in [3], where formulae were constrained to satisfy a well-formedness requirement, a restriction needed to characterise hereditary h-bisimilarity. In the next section, we will see that the logic where such restriction is dropped is expressive enough to capture isomorphism of PESs. Then, we shall extract out from this logic a number of sub-logics able to characterise all the equivalences presented in Section 1.

Definition 13 *(logic $\mathcal{L}$).* Let $\mathcal{V}$ be a denumerable set of variables ranged over by $x, y, z, \ldots$. The syntax of formulae of logic $\mathcal{L}$ is defined as follows, where a ranges over $\mathcal{A}$ and $\boldsymbol{x}$ and $\boldsymbol{y}$ denote finite (and possibly empty) sequences of pairwise different variables:

$$\varphi \ ::= \ \mathsf{T} \ \mid \ \varphi \wedge \varphi \ \mid \ \neg\varphi \ \mid \ (\boldsymbol{x}, \overline{\boldsymbol{y}} < a\, z)\, \varphi \ \mid \ \langle z \rangle\, \varphi$$

Formulae that do not use the binder $(\boldsymbol{x}, \overline{\boldsymbol{y}} < a\, z)$ are called *execution-only* and denoted by ξ.

Intuitively, the formula $(\boldsymbol{x}, \overline{\boldsymbol{y}} < a\, z)\, \varphi$ holds in a configuration when in the future of the configuration there is an a-labelled event e, consistent with the events bound to free variables in φ, such that, binding e to variable z, the formula φ holds. Such an event is required to be caused (at least) by the events already bound to variables in $\boldsymbol{x}$, and to be concurrent (at least) with those bound to variables in $\boldsymbol{y}$. We stress that the event e might not be currently enabled; it is only required to be consistent with the current configuration, meaning that it could be enabled in the future of the current configuration. The formula $\langle z \rangle\, \varphi$ says that the event bound to z is either enabled by the current configuration and its execution produces a new configuration which satisfies the formula φ, or it already belongs to the current configuration that itself satisfies the formula φ. To simplify notations, we write $(\boldsymbol{x} < a\, z)\, \varphi$ for $(\boldsymbol{x}, \overline{\varnothing} < a\, z)\, \varphi$, $(\overline{\boldsymbol{y}} < a\, z)\, \varphi$ for $(\varnothing, \overline{\boldsymbol{y}} < a\, z)\, \varphi$, and $(a\, z)\, \varphi$ for $(\varnothing, \overline{\varnothing} < a\, z)\, \varphi$.

The operator $(\boldsymbol{x}, \overline{\boldsymbol{y}} < a\, z)$ acts as a binder for the variable z, as clarified by the following notion of free variables in a formula.

Definition 14 *(free variables).* The set of free variables of a formula φ, denoted $fv(\varphi)$, is inductively defined by:

$$\begin{aligned}
fv(\mathsf{T}) &= \varnothing \\
fv(\varphi_1 \wedge \varphi_2) &= fv(\varphi_1) \cup fv(\varphi_2) \\
fv(\neg\varphi) &= fv(\varphi) \\
fv((\boldsymbol{x}, \overline{\boldsymbol{y}} < a\, z)\, \varphi) &= \boldsymbol{x} \cup \boldsymbol{y} \cup (fv(\varphi) \setminus \{z\}) \\
fv(\langle z \rangle\, \varphi) &= fv(\varphi) \cup \{z\}
\end{aligned}$$

The satisfaction of a formula φ is defined with respect to a configuration $C \in \mathit{Conf}(\mathcal{E})$, representing the state of the computation, and a (total) function $\eta : \mathcal{V} \to E$, called an *environment*, that binds free variables in φ to events in C or in the future of C. To this aim, given a configuration C, we denote by $E[C]$ the *residual* of E after C, defined as $E[C] = \{e \in E \setminus C \text{ s.t. } C \frown e\}$. We denote by $\mathit{Env}_{\mathcal{E}}$ the set of all environments $\eta : \mathcal{V} \to E$. Furthermore, given $\eta \in \mathit{Env}_{\mathcal{E}}$ and $e \in E$, we denote by $\eta[z \mapsto e]$ the element of $\mathit{Env}_{\mathcal{E}}$ such that

$$(\eta[z \mapsto e])(x) = \begin{cases} e & \text{if } x = z \\ \eta(x) & \text{otherwise} \end{cases}$$

Definition 15 *(semantics of $\mathcal{L}$).* Let $\mathcal{E}$ be a PES. The denotation in $\mathcal{E}$ of a formula $\varphi \in \mathcal{L}$, written $\{\!|\varphi|\!\}^{\mathcal{E}}$, is the subset of $Conf(\mathcal{E}) \times Env_{\mathcal{E}}$ inductively defined as follows:

$$\begin{aligned}
\{\!|\mathsf{T}|\!\}^{\mathcal{E}} &= Conf(\mathcal{E}) \times Env_{\mathcal{E}} \\
\{\!|\varphi_1 \wedge \varphi_2|\!\}^{\mathcal{E}} &= \{\!|\varphi_1|\!\}^{\mathcal{E}} \cap \{\!|\varphi_2|\!\}^{\mathcal{E}} \\
\{\!|\neg\varphi|\!\}^{\mathcal{E}} &= (Conf(\mathcal{E}) \times Env_{\mathcal{E}}) \setminus \{\!|\varphi|\!\}^{\mathcal{E}} \\
\{\!|(\boldsymbol{x}, \overline{\boldsymbol{y}} < a\, z)\, \varphi|\!\}^{\mathcal{E}} &= \{(C, \eta) \mid \exists e \in E[C] \ : \ l(e) = a \ \wedge \ \eta(\boldsymbol{x}) \leq e \ \wedge \ \eta(\boldsymbol{y}) \parallel e \\
&\qquad\qquad\qquad \wedge \ (C, \eta[z \mapsto e]) \in \{\!|\varphi|\!\}^{\mathcal{E}} \qquad \} \\
\{\!|\langle z \rangle\ \varphi|\!\}^{\mathcal{E}} &= \{(C, \eta) \mid C' = C \cup \{\eta(z)\} \in Conf(\mathcal{E}) \wedge (C', \eta) \in \{\!|\varphi|\!\}^{\mathcal{E}}\}
\end{aligned}$$

When $(C, \eta) \in \{\!|\varphi|\!\}^{\mathcal{E}}$, we say that the $\mathcal{E}$ satisfies φ in configuration C and environment η, and write $\mathcal{E}, C \models_{\eta} \varphi$. For closed formulae φ, we write $\mathcal{E}, C \models \varphi$ whenever $\mathcal{E}, C \models_{\eta} \varphi$ for some η, and $\mathcal{E} \models \varphi$ whenever $\mathcal{E}, \varnothing \models \varphi$; in this case, we say that *$\mathcal{E}$ satisfies φ.*

It is worth noticing that the semantics of the binding operator does not prevent from choosing for z an event e that has already been bound to a different variable, i.e., the environment function η need not be injective. Furthermore, in the semantics of $\langle z \rangle\, \varphi$, observe that the condition $C' = C \cup \{\eta(z)\} \in Conf(\mathcal{E})$ can be satisfied in two different ways. The first possibility is that $C \xrightarrow{l(\eta(z))} C'$, i.e., in C the event $\eta(z)$ is enabled and can be executed leading to $C' = C \cup \{\eta(z)\}$. Alternatively, it can be that $\eta(z) \in C$, i.e., the event $\eta(z)$ has been already executed in the past, and thus $C = C'$. This is a difference with respect to the original semantics in [3]. We will see that it does not alter the expressiveness of the fragments considered in [3], while allowing for a precise characterisation of the expressiveness of the full logic.

2.1. Derived operators

In what follows, we shall freely use disjunctions and F, obtained as usual from conjuctions and T by means of negations. We next provide a few derived operators that, on the one hand, clarify what can be expressed in the logic and, on the other hand, will be useful to define the fragments of the logic that correspond to the behavioural equivalences presented for PESs.

Executing a set of events Given a finite set of variables $X \subseteq \mathcal{V}$, we write $\langle X \rangle\ \varphi$ for the formula inductively defined by

- $\langle \varnothing \rangle\ \varphi \triangleq \varphi$; and
- $\langle X \rangle\ \varphi \triangleq \bigvee_{z \in X} \langle z \rangle \langle X \setminus \{z\} \rangle\, \varphi$, when $X \neq \varnothing$.

Intuitively, $\langle X \rangle\ \varphi$ states that from the current state, the events bound to the variables in X either already belong to the current configuration or can be executed, in some order, and then φ holds.

Immediate execution We write

$$\langle\!| \boldsymbol{x}, \overline{\boldsymbol{y}} < a\, z |\!\rangle\, \varphi \quad \text{for the formula} \quad (\boldsymbol{x}, \overline{\boldsymbol{y}} < a\, z) \langle z \rangle\, \varphi$$

that states the existence of an event e enabled by the current configuration, and thus which can be immediately executed, such that after executing e the formula φ holds (with e bound to variable z). Notice that the fact that $\langle z \rangle$ is preceded by $(\boldsymbol{x}, \overline{\boldsymbol{y}} < a\, z)$ excludes the possibility of satisfying $\langle z \rangle$ by associating to z an event that is already in the current configuration.

Steps We introduce a notation also to predicate the existence (resp., the immediate execution) of concurrent events. We write

$$\begin{aligned}
((a\, z) \otimes (b\, z'))\, \varphi &\quad \text{for the formula} \quad (a\, z)(\overline{z} < b\, z')\varphi \\
(\langle\!| a\, z |\!\rangle \otimes \langle\!| b\, z' |\!\rangle)\varphi &\quad \text{for the formula} \quad ((a\, z) \otimes (b\, z'))\langle z \rangle\ \langle z' \rangle\, \varphi
\end{aligned}$$

The first formula declares the existence of two concurrent events, labelled by a and b, respectively, such that, if we bind such events to z and z', then φ holds. The second formula states the existence of two concurrently enabled events, labelled by a and b, whose immediate execution leads to a state where φ holds.

Clearly, this notation can be generalized to the quantification and the immediate execution of any number of concurrent events.

Well-formed quantification Later it will be convenient to restrict the use of the quantification $(\boldsymbol{x}, \overline{\boldsymbol{y}} < a\,z)\,\varphi$, by requiring that the relation of z with each variable which occurs free in φ is specified.

Definition 16 *(well-formedness).* A quantification $(\boldsymbol{x}, \overline{\boldsymbol{y}} < a\,z)\,\varphi$ is *well-formed* if $fv(\varphi) \subseteq \boldsymbol{x} \cup \boldsymbol{y} \cup \{z\}$.

In order to ease the writing of formulae, whenever we are interested in specifying the relation of z only with some of the free variables of φ, we introduce the following notation:

$$((\boldsymbol{x}, \overline{\boldsymbol{y}} < a\,z))\,\varphi \;\triangleq\; \bigvee_{\substack{\boldsymbol{v} \subseteq fv(\varphi)\setminus(z\boldsymbol{xy}) \\ \boldsymbol{w} = fv(\varphi)\setminus(z\boldsymbol{xyv})}} (\boldsymbol{xv}, \overline{\boldsymbol{yw}} < a\,z)\,\varphi$$

In words, for those variables in $fv(\varphi)$ whose relation with z is not specified, we consider all possibilities of causes and concurrent events by taking all the possible bipartitions (in the sets $\boldsymbol{v}$ and $\boldsymbol{w}$). Notice that this maintains well-formedness.

Executability check Let X be a (finite) set of variables. We let

$$(a\,z\rangle_X\,\varphi \;\triangleq\; ((a\,z))\;(\langle X\rangle\,\langle z\rangle\,\mathsf{T} \wedge \varphi)$$

Intuitively, $(a\,z\rangle_X\,\varphi$ states that there is an a-labelled event that could be executed after the events in X and, if we bind such event to z *without executing it*, the formula φ holds.

3. Logical equivalences

The work in [3] shows that a number of behavioural equivalences in the true concurrent spectrum can be logically characterised in terms of suitable fragments of the logic $\mathcal{L}$ presented in the previous section. More specifically, the logic characterising hereditary h-bisimilarity is $\mathcal{L}$ restricted to well-formed formulae. In this section, we first show that the logic without well-formation characterises isomorphism of event structures. Moreover, we complete the spectrum by characterising various forms of trace equivalences and identifying fragments of the logic that capture weak h-bisimilarities, two behavioural equivalences which escaped the characterisations in [3].

In what follows, we will often refer to the notion of logical equivalence with respect to some fragment of the logic.

Definition 17 *(logical equivalence).* Let $\mathcal{L}'$ be some fragment of the logic $\mathcal{L}$. We say that two PESs $\mathcal{E}_1$ and $\mathcal{E}_2$ are *logically equivalent* with respect to $\mathcal{L}'$, written $\mathcal{E}_1 \equiv_{\mathcal{L}'} \mathcal{E}_2$, whenever they satisfy the same closed formulae of $\mathcal{L}'$.

3.1. Logical characterisation of the spectrum: what is known from [3]

We start with a recap of the results in [3]. As mentioned before, logic $\mathcal{L}$ in full generality results to be too expressive to characterise hereditary h-bisimilarity. For having $\approx_{\text{hhb}}$ as logical equivalence, we must consider a fragment where quantifications are well-formed, as defined in the previous section. Equivalently, one could restrict the semantics to the so-called *legal pairs*, but here (differently from [3]) we favour the syntactic approach.

Definition 18 *(fragments of $\mathcal{L}$ corresponding to various behavioural equivalences).*

Interleaving Logic ($\mathcal{L}_i$)	$\varphi \;::=\; \langle\!|a\,x|\!\rangle\,\varphi \;\mid\; \varphi \wedge \varphi \;\mid\; \neg\varphi \;\mid\; \mathsf{T}$
Step Logic ($\mathcal{L}_s$)	$\varphi \;::=\; (\langle\!|a_1\,x_1|\!\rangle \otimes \cdots \otimes \langle\!|a_n\,x_n|\!\rangle)\;\varphi \;\mid\; \varphi \wedge \varphi \;\mid\; \neg\varphi \;\mid\; \mathsf{T}$
Pomset Logic ($\mathcal{L}_p$)	$\varphi \;::=\; \langle\!|\boldsymbol{x}, \overline{\boldsymbol{y}} < a\,z|\!\rangle\,\varphi \;\mid\; \varphi \wedge \varphi \;\mid\; \neg\varphi \;\mid\; \mathsf{T}$ where $\neg$ and $\wedge$ are used only on closed formulae.
History Preserving Logic ($\mathcal{L}_h$)	$\varphi \;::=\; \langle\!|\boldsymbol{x}, \overline{\boldsymbol{y}} < a\,z|\!\rangle\,\varphi \;\mid\; \varphi \wedge \varphi \;\mid\; \neg\varphi \;\mid\; \mathsf{T}$
Hereditary History Preserving Logic ($\mathcal{L}_{hh}$)	$\mathcal{L}$ where all quantifications are well-formed

It has been shown that the above fragments provide a logical characterisation of interleaving, step, pomset, history and hereditary history preserving bisimilarities.

Theorem 1 *(logical characterisation of bisimilarities [3]). Let $\mathcal{E}_1$ and $\mathcal{E}_2$ be two PESs. Then, $\mathcal{E}_1 \equiv_{\mathcal{L}_\alpha} \mathcal{E}_2$ if and only if $\mathcal{E}_1 \approx_{\alpha\mathrm{b}} \mathcal{E}_2$, for every $\alpha \in \{\mathrm{i}, \mathrm{s}, \mathrm{p}, \mathrm{h}, \mathrm{hh}\}$.*

Recall that the semantics of operator $\langle z \rangle\, \varphi$ in this paper is slightly different from the one in [3], where the formula is false if the event bound to z has been already executed. However, it is trivial to see that this does not affect the result above. In fact, in all fragments except $\mathcal{L}_{hh}$, an event is always executed immediately after being quantified, hence it is not possible to execute the same event twice. Instead, in the logic $\mathcal{L}_{hh}$, it is not possible to bind the same event to different variables and later use them in execution modalities. In fact, by the well-formedness condition, the event bound to the second variable will be either concurrent or causally dependent on the event bound to first one and thus necessarily distinct. Therefore, the modification to the semantics plays a real role only in the full logic $\mathcal{L}$.

3.2. Logical equivalence for $\mathcal{L}$ is isomorphism

We next show that the logic $\mathcal{L}$ is expressive enough to tell apart any two PESs which are not isomorphic. Intuitively, without the well-formedness requirement, we can reason on events which are in conflict; for instance, this could allow us to distinguish the PES consisting of a single a-labelled event from that consisting of two conflicting a-labelled events. In fact, the second PES satisfies $(a\, x)(a\, y)\langle x \rangle \neg\langle y \rangle \mathsf{T}$, while the first one does not, since x and y must be bound to the unique event. This suggests that, for PESs, hereditary h-bisimilarity essentially differs from isomorphism only for the possibility of merging conflicting equivalent branches, i.e., for the addition of an axiom of the kind $E + E = E$; this is similar to what happens in other frameworks, as proved e.g. in [23] for BPP. A further evidence of this fact is that $\approx_{\mathrm{hhb}}$ and $\cong$ remain different when causality is removed (i.e., in the setting of CSs), whereas they collapse as soon as conflict is removed (i.e., in the setting of EESs): see Figg. 11 and 12.

In order to prove the desired result, the idea is to show that, under the assumption of image-finiteness, a PES can be approximated by means of suitably defined finite prefixes. Observe that the idea of considering prefixes up to some fixed causal depth would not work, since these might be infinite. For instance, consider again the PES $\mathcal{E}$ with events $E = \{e_i \mid i \in \mathbb{N}\}$, empty conflict and causality and labelling $l(e_i) = a_i$, where all a_i are distinct labels. We observed that $\mathcal{E}$ is image-finite but all its infinitely many events have causal depth 0.

We proceed as follows. We fix some enumeration of the set of labels $\lambda : \mathcal{A} \to \mathbb{N}$. Then, we define the level of an event as a suitable combination of the causal depth and the position of the label in the enumeration. More precisely, for an event $e \in E$, its *level* is inductively defined as $lev(e) = \max\big(\{lev(e') + 1 \mid e' \in E \setminus \{e\} \ \wedge e' \leq e\} \cup \{\lambda(l(e))\}\big)$, where it is intended that $\max(\emptyset) = 0$.

Definition 19 *(k-prefix of a PES).* Let $\mathcal{E}$ be a PES. For $k \in \mathbb{N}$, let $E^{(k)} = \{e \mid e \in E \wedge lev(e) \leq k\}$ be the set of events of $\mathcal{E}$ whose level is at most k. Then, the *k-prefix* of $\mathcal{E}$ is the PES defined as $\mathcal{E}^{(k)} = (E^{(k)}, \leq|_{E^{(k)}}, \#|_{E^{(k)}}, l|_{E^{(k)}})$.

Note that, by the very definition of level, $lev(e') \leq lev(e)$ for all $e' \leq e$; hence, the k-prefix $E^{(k)}$ of a PES $\mathcal{E}$ is a causally closed subset of $\mathcal{E}$. From this observation, it immediately follows that $\mathcal{E}^{(k)}$ is indeed a PES, i.e., the definition is well-given.

While in general the k-prefix of a PES could be infinite because the PES could be infinite in width, this cannot happen for image-finite PESs.

Lemma 1 *(finiteness of k-prefix). Let $\mathcal{E}$ be an image-finite PES. For all $k \in \mathbb{N}$, $E^{(k)}$ is finite, hence $\mathcal{E}^{(k)}$ is finite.*

Proof. In order to simplify the notation, denote by a_i the label in $\mathcal{A}$ such that $\lambda(a_i) = i$.

We proceed by induction on k.

($k = 0$) Just note that $\mathcal{E}^{(0)}$ includes events labelled by a_0 and with empty set of causes. These are all enabled at the empty configuration and thus they are finitely many by image-finiteness.

($k \to k+1$) Observe that, for each event e in $\mathcal{E}^{(k+1)} \setminus \mathcal{E}^{(k)}$, by definition, $lev(e) = k+1$. Therefore, $l(e) = a_j$ with $j \leq k+1$. Moreover, if we consider $\lfloor e \rfloor$, i.e. the set of causes of e, necessarily $\lfloor e \rfloor \subseteq \mathcal{E}^{(k)}$. This means that e is enabled by some configuration in $\mathcal{E}^{(k)}$. Summing up, events in $\mathcal{E}^{(k+1)} \setminus \mathcal{E}^{(k)}$ are enabled by some of the finitely many configurations of $\mathcal{E}^{(k)}$ and are labelled by one of $a_0, \ldots, a_{k+1}$. Therefore, they are necessarily finite, otherwise there would be at least one configuration in $\mathcal{E}^{(k)}$ enabling infinitely many events with the same label. □

A simple but crucial observation is that non-isomorphic image-finite PESs can be distinguished at some finite level. This is expressed by the following lemma.

Lemma 2 *(finite distinguishability). Let $\mathcal{E}_1$ and $\mathcal{E}_2$ be two image-finite PESs. Then $\mathcal{E}_1 \cong \mathcal{E}_2$ iff $\mathcal{E}_1^{(k)} \cong \mathcal{E}_2^{(k)}$, for all $k \in \mathbb{N}$.*

Proof. ($\Rightarrow$) This is immediate, since an isomorphism $f : \mathcal{E}_1 \to \mathcal{E}_2$ restricts to an isomorphism $f^{(k)} : \mathcal{E}_1^{(k)} \to \mathcal{E}_2^{(k)}$, for every k.

($\Leftarrow$) Assume that, for each k, the prefixes $\mathcal{E}_1^{(k)}$ and $\mathcal{E}_2^{(k)}$ are isomorphic. If $\mathcal{E}_1^{(k)}$ is finite, we trivially conclude. Otherwise, consider the set $I = \bigcup_{k\in\mathbb{N}} \{f^{(k)} \mid f^{(k)} : \mathcal{E}_1^{(k)} \to \mathcal{E}_2^{(k)}$ isomorphism$\}$. Ordered by subset inclusion, I is a tree: the root is the only isomorphism $f^{(0)}$ between the empty structures; each $f^{(k)}$, with $k > 0$, has a unique predecessor, i.e., $f^{(k)}|_{\mathcal{E}_i^{(k-1)}}$. Furthermore, it is finitely branching, since each $\mathcal{E}_i^{(k)}$ is finite, and it is clearly infinite, since $\mathcal{E}_1$ and $\mathcal{E}_2$ are. Therefore, it has an infinite branch, say $(f^{(k)})_{k\in\mathbb{N}}$. It is easy to see that $f = \bigcup_{k\in\mathbb{N}} f^{(k)} : \mathcal{E}_1 \to \mathcal{E}_2$ is an isomorphism. □

The next crucial observation is that we can (almost) completely characterise a finite PES via a formula. Here and in what follows, given a PES $\mathcal{E}$ with n events, we denote with $lin(E)$ the set of all *linearisations* of E, i.e., the set of all sequences $e_1, \dots, e_n$ such that, for all $i, j \in \{1, \dots, n\}$, if $e_i \leq_E e_j$ then $i \leq j$.

Definition 20 *(characteristic formula).* Let $\mathcal{E}$ be a finite PES and $e_1, \dots, e_n \in lin(E)$. Let $l(e_i) = a_i$ and consider a set of variables $X = \{x_1, \dots, x_n\}$. The *characteristic formula* of $\mathcal{E}$, denoted by $\chi(\mathcal{E})$, is defined by

$$(a_1\, x_1)\,(x_2^<, x_2^\| < a_2\, x_2) \dots (x_n^<, x_n^\| < a_n\, x_n) \left(\bigwedge_{x_i \in X} \langle x_i^< \rangle \langle x_i \rangle \mathsf{T} \quad \wedge \quad \bigwedge_{e_i \# e_j} \bigwedge_{X' \subseteq X} \neg \langle X' \cup \{x_i, x_j\} \rangle \mathsf{T} \right)$$

where $x_j^< \triangleq \{x_i : i < j \ \wedge \ e_i \leq e_j\}$ and $x_j^\| \triangleq \{x_i : i < j \ \wedge \ e_i \parallel e_j\}$.

In words, the characteristic formula asks for the existence of events, bound to $x_1, \dots, x_n$. The quantification of each x_i requires the corresponding event to be caused by events in $x_i^<$ and concurrent to events in $x_i^\|$, mimicking what happens in the given event structure. Asking that $\langle x_i^< \rangle \langle x_i \rangle \mathsf{T}$ holds, imposes that the set of causes of x_i is exactly $x_i^<$ (otherwise it could be larger). Finally, since all events have causes in X, asking that $\bigwedge_{e_i \# e_j} \bigwedge_{X' \subseteq X} \neg \langle X' \cup \{x_i, x_j\} \rangle \mathsf{T}$ holds amounts to asking that the events bound to x_i and x_j are in conflict, whenever $e_i \# e_j$.

The characteristic formula almost entirely characterises a finite event structure, in the sense formalised below.

Lemma 3 *(rigid embedding). Let $\mathcal{E}$ and $\mathcal{E}'$ be two PESs, with $\mathcal{E}$ finite. Then $\mathcal{E}' \models \chi(\mathcal{E})$ iff there is a rigid embedding of $\mathcal{E}$ into $\mathcal{E}'$, i.e., an injective map $f : \mathcal{E} \to \mathcal{E}'$ such that, for all $e_1, e_2 \in \mathcal{E}$, (i) $f(\lfloor e_1 \rfloor) = \lfloor f(e_1) \rfloor$ and (ii) $e_1 \# e_2$ if and only if $f(e_1) \# f(e_2)$.*

Proof. ($\Rightarrow$) Let $\mathcal{E}$ be a finite PES, $(e_1, \dots, e_n) \in lin(E)$, $l(e_i) = a_i$, and $X = \{x_1, \dots, x_n\}$ be the variables used in the characteristic formula $\chi(\mathcal{E})$. Since $\mathcal{E}' \models \chi(\mathcal{E})$, it must be

$$\mathcal{E}', \varnothing \models_\eta \bigwedge_{x_i \in X} \langle x_i^< \rangle \langle x_i \rangle \mathsf{T} \ \wedge \ \bigwedge_{e_i \# e_j} \bigwedge_{X' \subseteq X} \neg \langle X' \cup \{x_i, x_j\} \rangle \mathsf{T}$$

for a suitable environment $\eta : \mathcal{V} \to E'$. If we let $\eta(x_i) = e'_i$, then by the shape of the quantifiers that bind $x_1, \dots, x_n$, for all i, j with $i < j$, if $e_i \leq e_j$ then $e'_i \leq e'_j$, and if $e_i \parallel e_j$ then $e'_i \parallel e'_j$. Since the first conjunct $\bigwedge_{x_i \in X} \langle x_i^< \rangle \langle x_i \rangle \mathsf{T}$ is satisfied, all events e'_i have causes in $\{e'_j \mid j < i\}$. Since the second conjunct $\bigwedge_{e_i \# e_j} \bigwedge_{X' \subseteq X} \neg \langle X' \cup \{x_i, x_j\} \rangle \mathsf{T}$ is satisfied, if $e_i \# e_j$, then $e'_i \# e'_j$, since they are not executable in the same computation.

Now, if we define a mapping $f : \mathcal{E} \to \mathcal{E}'$ by $f(e_i) = e'_i$, for $i \in \{1, \dots, n\}$, we conclude that f is a rigid embedding of $\mathcal{E}$ into $\mathcal{E}'$. Note that f is indeed injective, since, if $e_i \neq e_j$, then the two events are either causally dependent or concurrent or in conflict; the same must hold for $f(e_i) = e'_i$ and $f(e_j) = e'_j$, which are thus different.

($\Leftarrow$) Let $\chi(\mathcal{E})$ be the characteristic formula of $\mathcal{E}$, as in Definition 20, and let $e_1, \dots, e_n$ be the linearisation of $\mathcal{E}$ used to construct the formula. Consider the events $f(e_1), \dots, f(e_n)$ in $\mathcal{E}'$. Since $f : \mathcal{E} \to \mathcal{E}'$ is a rigid embedding, for all $i, j \in \{1, \dots, n\}$ we have:

- $e_i \leq e_j$ iff $f(e_i) \leq f(e_j)$ and $\lfloor f(e_i) \rfloor = \{f(e_k) \mid k \leq i \ \wedge \ e_k \leq e_i\}$ (by property (i) of rigid embeddings)
- $e_i \# e_j$ iff $f(e_i) \# f(e_j)$ (by property (ii) of rigid embeddings)

Therefore, we conclude that $\mathcal{E}' \models \chi(\mathcal{E})$, by binding variables $x_1, \dots, x_n$ to $f(e_1), \dots, f(e_n)$. □

Lemma 4 *(k-prefixes, logically). Let $\mathcal{E}$ be an image-finite PES. For every $k \in \mathbb{N}$, $\mathcal{E}$ satisfies the characteristic formula of $\mathcal{E}^{(k)}$.*

Proof. Just observe that the inclusion of $\mathcal{E}^{(k)}$ into $\mathcal{E}$ is a rigid embedding, and exploit Lemma 3. □

Theorem 2 *(logical characterisation of isomorphism). Let $\mathcal{E}_1$ and $\mathcal{E}_2$ be two image-finite PESs. Then, $\mathcal{E}_1 \equiv_{\mathcal{L}} \mathcal{E}_2$ if and only if $\mathcal{E}_1 \cong \mathcal{E}_2$.*

Proof. $(\Rightarrow)$. Assume that $\mathcal{E}_1 \equiv_{\mathcal{L}} \mathcal{E}_2$. Then, for all $k \in \mathbb{N}$, consider the characteristic formula of the k-prefix, $\chi(\mathcal{E}_1^{(k)})$. By Lemma 4, we know that $\mathcal{E}_1 \models \chi(\mathcal{E}_1^{(k)})$. Since $\mathcal{E}_1 \equiv_{\mathcal{L}} \mathcal{E}_2$, this implies $\mathcal{E}_2 \models \chi(\mathcal{E}_1^{(k)})$ and thus, by Lemma 3, there is a rigid embedding $f : \mathcal{E}_1^{(k)} \to \mathcal{E}_2$. Since a rigid embedding clearly preserves the causal depth of the events, i.e., $d(f(e_1)) = d(e_1)$ for all e_1 in $\mathcal{E}_1^{(k)}$, it can be seen as a rigid embedding into the k-prefix of $\mathcal{E}_2$, i.e., $f^{(k)} : \mathcal{E}_1^{(k)} \to \mathcal{E}_2^{(k)}$. A symmetric argument shows that there is a rigid embedding $g^{(k)} : \mathcal{E}_2^{(k)} \to \mathcal{E}_1^{(k)}$. Since k-prefixes are finite, this implies that $f^{(k)} \circ g^{(k)}$ and $g^{(k)} \circ f^{(k)}$ are automorphisms. Therefore, $f^{(k)} : \mathcal{E}_1^{(k)} \to \mathcal{E}_2^{(k)}$ is an isomorphism whose inverse is $(g^{(k)} \circ (f^{(k)} \circ g^{(k)})^{-1})$ and thus $\mathcal{E}_1^{(k)} \cong \mathcal{E}_2^{(k)}$, as desired. Since this holds for all k, by Lemma 2 we conclude that $\mathcal{E}_1 \cong \mathcal{E}_2$, as desired.

$(\Leftarrow)$ Immediate. $\square$

3.3. Completing the spectrum

In this section we discuss how to complete the characterisation of equivalences in the true concurrent spectrum. This requires including trace equivalences and weak h-bisimilarities.

Trace equivalences Logics characterising trace equivalences can be obtained from the fragments for corresponding bisimilarities, omitting conjunction and negation, but keeping F (that is, $\neg\mathsf{T}$). Adding F does not change the logical equivalence induced, since such a formula is not satisfied by any PESs; however, it will allow for a simple encoding of one logic into another in Section 4.1.

Definition 21 *(logics for trace equivalences).* For $\alpha \in \{\mathrm{i}, \mathrm{s}, \mathrm{p}\}$, we write $\mathcal{L}_\alpha^t$ to denote the logics obtained from $\mathcal{L}_\alpha$ by removing the operators $\wedge$ and $\neg$, and by adding F.

In order to get to the desired result, we first recall from [3] how to build a $\mathcal{L}_p$ formula associated to a given pomset. Given a tuple of variables $\boldsymbol{z} = z_1, \dots, z_n$ and a tuple of labels $\boldsymbol{a} = a_1, \dots, a_n$, we write $\boldsymbol{az}$ to indicate a tuple of labelled variables.

Definition 22 *(pomsets as formulae in $\mathcal{L}_p$).* Let $\boldsymbol{az} = a_1 z_1, \dots, a_n z_n$ be a tuple of labelled variables and consider the poset $p_{\boldsymbol{az}} = (\boldsymbol{az}, <)$, where $<$ is a strict order on $\boldsymbol{z}$ compatible with listing $\boldsymbol{z}$ (i.e., if $z_i < z_j$ then $i < j$).

Given a formula $\varphi \in \mathcal{L}_p$, we denote by $\langle\!|\, p_{\boldsymbol{az}} |\!\rangle\varphi$ the formula inductively defined as follows:

- If $\boldsymbol{z}$ is empty, then $\langle\!|\, p_{\boldsymbol{az}} |\!\rangle\varphi \triangleq \varphi$.
- If $\boldsymbol{az} = \boldsymbol{a}'\boldsymbol{z}', bw$, let $\boldsymbol{x} = \{z \in \boldsymbol{z}' \mid z < w\}$ and $\boldsymbol{y} = \boldsymbol{z}' \setminus \boldsymbol{x}$; then, $\langle\!|\, p_{\boldsymbol{az}} |\!\rangle\varphi \triangleq \langle\!|\, p_{\boldsymbol{a}'\boldsymbol{z}'} |\!\rangle \, \langle\!| \boldsymbol{x}, \overline{\boldsymbol{y}} < b\, w |\!\rangle\varphi$.

Then, it is immediate to show the following:

Lemma 5 *(transitions, logically). Let $\mathcal{E}$ be a PES and let $C \in Conf(\mathcal{E})$ be a configuration. Then, $C \xrightarrow{a} C'$ iff $\mathcal{E}, C \models \langle\!| a\, x |\!\rangle\mathsf{T}$. Similarly, for a multiset $\{\!| a_1 \dots a_n |\!\}$, we have $C \xrightarrow{\{\!| a_1 \dots a_n |\!\}} C'$ iff $\mathcal{E}, C \models (\langle\!| a_1\, x_1 |\!\rangle \otimes \dots \otimes \langle\!| a_n\, x_n |\!\rangle)\mathsf{T}$. Finally, for a labelled poset $p_{\boldsymbol{az}} = (\boldsymbol{az}, <)$, we have $C \xrightarrow{pomset(p_{\boldsymbol{az}})} C'$ iff $\mathcal{E}, C \models_\eta \langle\!|\, p_{\boldsymbol{az}} |\!\rangle\mathsf{T}$.*

Proof. The claim for interleaving and step holds by the semantics of $\mathcal{L}$, whereas the claim for pomset holds by Lemma 5.3 and 5.5 from [3]. $\square$

Proposition 1 *(logical characterisation of trace equivalences). Let $\mathcal{E}_1$ and $\mathcal{E}_2$ be two PESs. Then, $\mathcal{E}_1 \equiv_{\mathcal{L}_\alpha^t} \mathcal{E}_2$ if and only if $\mathcal{E}_1 \approx_{\alpha\mathrm{t}} \mathcal{E}_2$, for every $\alpha \in \{i, s, p\}$.*

Proof. Given an interleaving/step trace t, we let $\langle\!|\, t |\!\rangle$ denote

- $\langle\!| a_1\, x_1 |\!\rangle \dots \langle\!| a_n\, x_n |\!\rangle$ whenever $t = a_1 \dots a_n$;
- $(\langle\!| a_1^1\, x_1^1 |\!\rangle \otimes \dots \otimes \langle\!| a_{m_1}^1\, x_{m_1}^1 |\!\rangle) \dots (\langle\!| a_1^n\, x_1^n |\!\rangle \otimes \dots \otimes \langle\!| a_{m_n}^n\, x_{m_n}^n |\!\rangle)$ whenever $t = \{\!| a_1^1, \dots, a_{m_1}^1 |\!\} \dots \{\!| a_1^n, \dots, a_{m_n}^n |\!\}$.

$(\Rightarrow)$ First, suppose that t belongs to $SeqTr(\mathcal{E}_1)$, $StepTr(\mathcal{E}_1)$ or $Pom(\mathcal{E}_1)$, respectively; by using Lemma 5, we have that $\mathcal{E}_1 \models \langle\!|\, t |\!\rangle\mathsf{T}$. By hypothesis, $\mathcal{E}_2 \models \langle\!|\, t |\!\rangle\mathsf{T}$ that, by Definition 15, implies that t belongs to $SeqTr(\mathcal{E}_2)$, $StepTr(\mathcal{E}_2)$ or $Pom(\mathcal{E}_2)$, respectively.

$(\Leftarrow)$ Let $\varphi \in \mathcal{L}_\alpha^t$ such that $\mathcal{E}_1 \models \varphi$. For $\alpha = i$, we have that $\varphi = \langle\!| a_1\, x_1 |\!\rangle \dots \langle\!| a_n\, x_n |\!\rangle\mathsf{T}$, for some n and $a_1, x_1, \dots, a_n, x_n$; by Lemma 5, $a_1 \dots a_n \in SeqTr(\mathcal{E}_1)$. By hypothesis, $a_1 \dots a_n \in SeqTr(\mathcal{E}_2)$ and so, again by Lemma 5, $\mathcal{E}_2 \models \varphi$. A similar reasoning can be done for $\alpha = s$, with $(\langle\!| a_1^i\, x_1^i |\!\rangle \otimes \dots \otimes \langle\!| a_{m_i}^i\, x_{m_i}^i |\!\rangle)$ in place of $\langle\!| a_i\, x_i |\!\rangle$ and $StepTr(\cdot)$ in place of $SeqTr(\cdot)$. For $\alpha = p$, we only have to prove that every formula of $\mathcal{L}_p^t$ satisfied by $\mathcal{E}$ denotes a pomset of $\mathcal{E}$; this is Lemma 5.3 of [3]. Then, the reasoning is like before, by using $Pom(\cdot)$ in place of $StepTr(\cdot)$ and $SeqTr(\cdot)$. $\square$

Weak history preserving bisimilarities We next identify fragments of $\mathcal{L}$ that correspond to weak and weak pomset h-bisimilarities. We start with the logic for weak h-bisimilarity.

Definition 23 *(logic for weak h-bisimilarity).* Define, for a tuple of labelled variables $\boldsymbol{ax}$

$$\varphi_{\boldsymbol{ax}} ::= \mathsf{T} \mid \neg\,\varphi_{\boldsymbol{ax}} \mid \varphi_{\boldsymbol{ax}} \wedge \varphi_{\boldsymbol{ax}} \mid (b\,z\rangle_{\boldsymbol{x}}\,\varphi_{\boldsymbol{ax},bz} \mid \bigvee_{\substack{\sigma:\boldsymbol{ax}\to\boldsymbol{ax}\\ \textit{automorphism}}} \xi_{\boldsymbol{x}}\sigma$$

where it is intended that $\xi_{\boldsymbol{x}}$ is an execution-only formula (see Definition 13) with $fv(\xi_{\boldsymbol{x}}) \subseteq \boldsymbol{x}$, σ is a bijection over $\boldsymbol{x}$ which preserves the variable labels, and by $\xi_{\boldsymbol{x}}\sigma$ we denote the formula $\xi_{\boldsymbol{x}}$ where each variable $x \in \boldsymbol{x}$ is replaced by $\sigma(x)$. Then, $\mathcal{L}_{wh}$ is the set of formulae arising with $\boldsymbol{ax}$ empty, i.e., φ_{ϵ}.

We are now left with proving that $\mathcal{L}_{wh}$ characterises $\approx_{\text{whb}}$. This requires some ingenuity and so corresponding proofs are relegated to the Appendix. Here we just provide the auxiliary notions and results needed for the proof.

Definition 24 *(compatible environment).* Let $\mathcal{E}$ be a PES and $\boldsymbol{ax} = a_1x_1, \ldots, a_nx_n$ be a tuple of labelled variables. We say that an environment $\eta \in Env_{\mathcal{E}}$ is *$\boldsymbol{ax}$-compatible* when the following hold:

- $|\eta(\boldsymbol{x})| = |\boldsymbol{x}|$, i.e., η is injective on $\boldsymbol{x}$;
- $l(\eta(x_i)) = a_i$, for all $i \in \{1, \ldots, n\}$; and
- $\eta(\boldsymbol{x}) \in Conf(\mathcal{E})$.

Observe that, for some fixed tuple of labelled variables $\boldsymbol{ax}$ and $\boldsymbol{ax}$-compatible environment η, given a configuration $C \subseteq \eta(\boldsymbol{x})$, we have that $\mathcal{E}, C \models_{\eta} \langle \boldsymbol{x} \rangle\, \mathsf{T}$. Indeed, events in $\eta(\boldsymbol{x})$ can be executed in some order starting from the configuration C and, by definition of the semantics of the operator $\langle x \rangle\, \varphi$, events already executed are simply ignored.

In order to understand the logic for weak h-bisimilarity, recall that such behavioural equivalence can, in fact, be seen as an interleaving bisimulation that only relates isomorphic configurations. Indeed, the distinguishing power of weak h-bisimilarity is based on two distinct capabilities: the possibility of checking the executability of events with a specific label, and the possibility of verifying that the reached configurations are in the same equivalence class with respect to poset isomorphism. The syntax of the logic fragment reflects such dichotomy. On the one hand, the operator $(a\,z\rangle_X \varphi$ allows for checking the executability of a (labelled) action. This part induces a behavioural equivalence which is (at least) an interleaving bisimulation. On the other hand, the execution-only formulae $\xi_{\boldsymbol{x}}$ allow for testing whether the configuration identified by $\eta(\boldsymbol{x})$ belongs to a specific equivalence class of poset isomorphism; thus, only isomorphic configurations satisfy the same formulae. In order to make the naming of variables and the order in which they have been bound irrelevant and consider only their labels and causal relations, execution-only formulae are disjunctively applied to all possible renamings of variables preserving the labels. The strict separation between the two kinds of operators ensures that the logic is not more expressive than intended. The crucial intuition about execution-only formulae is formalised in the main auxiliary result below.

Lemma 6 *(invariance under poset isomorphism). Let $\mathcal{E}_1$ and $\mathcal{E}_2$ be two PESs. Given a tuple of labelled variables $\boldsymbol{ax}$ and two $\boldsymbol{ax}$-compatible environments $\eta_1 \in Env_{\mathcal{E}_1}$ and $\eta_2 \in Env_{\mathcal{E}_2}$, the following statements are equivalent:*

1. *$poset(\eta_1(\boldsymbol{x})) \cong poset(\eta_2(\boldsymbol{x}))$;*
2. *$\mathcal{E}_1, \varnothing \models_{\eta_1} \varphi_{\boldsymbol{ax}}$ if and only if $\mathcal{E}_2, \varnothing \models_{\eta_2} \varphi_{\boldsymbol{ax}}$, for every $\boldsymbol{ax}$-parameterised formula $\varphi_{\boldsymbol{ax}}$ of the form $\displaystyle\bigvee_{\substack{\sigma:\boldsymbol{ax}\to\boldsymbol{ax}\\ \textit{automorphism}}} \xi_{\boldsymbol{x}}\sigma$.*

Another simple but useful technical result is that the satisfaction of any $\boldsymbol{ax}$-parameterised formula $\varphi_{\boldsymbol{ax}}$ is independent from the naming of the variables in $\boldsymbol{x}$, as long as labels are preserved. Thus, given a compatible environment, one can apply any label preserving bijection to it, hence obtaining another compatible environment, without affecting the satisfaction of the formula.

Lemma 7 *(semi-permutable semantics). Let $\mathcal{E}$ be a PES, $\boldsymbol{ax} = a_1x_1, \ldots, a_nx_n$ be labelled variables and $\varphi_{\boldsymbol{ax}}$ be an $\boldsymbol{ax}$-parameterised formula as in Definition 23. Given an $\boldsymbol{ax}$-compatible environment $\eta \in Env_{\mathcal{E}}$ and a bijection $\sigma : \eta(\boldsymbol{x}) \to \eta(\boldsymbol{x})$ such that $\sigma \circ \eta$ is $\boldsymbol{ax}$-compatible, it holds that $\mathcal{E}, \varnothing \models_{\eta} \varphi_{\boldsymbol{ax}}$ if and only if $\mathcal{E}, \varnothing \models_{\sigma\circ\eta} \varphi_{\boldsymbol{ax}}$.*

Combining the facts above with the aforementioned intuition about the use of the executability check operator $(a\,z\rangle_X \varphi$, we obtain the final desired result.

Theorem 3 *(logical characterisation of $\approx_{\text{whb}}$). Let $\mathcal{E}_1$ and $\mathcal{E}_2$ be two image-finite PESs. Then, $\mathcal{E}_1 \equiv_{\mathcal{L}_{wh}} \mathcal{E}_2$ if and only if $\mathcal{E}_1 \approx_{\text{whb}} \mathcal{E}_2$.*

The logic for weak pomset h-bisimilarity can be obtained by slightly enlarging the fragment for weak h-bisimilarity with the possibility of checking for pomsets. More precisely, let us first adapt Definition 22, to quantification instead of executions.

Definition 25 *(pomset quantification).* Let $\boldsymbol{az} = a_1 z_1 \dots a_n z_n$ be a tuple of labelled variables and let $p_{\boldsymbol{az}} = (\boldsymbol{az}, <)$ be a poset with $<$ a strict order compatible with the listing $\boldsymbol{z}$ (i.e., if $z_i < z_j$ then $i < j$). Given a formula $\varphi \in \mathcal{L}$, we denote by $(\, p_{\boldsymbol{az}})\varphi$ the formula inductively defined as follows:

- If $\boldsymbol{z}$ is empty, then $(\, p_{\boldsymbol{az}})\varphi \triangleq \varphi$.
- If $\boldsymbol{az} = \boldsymbol{a}'\boldsymbol{z}', bw$, let $\boldsymbol{x} = \{z \in \boldsymbol{z}' \mid z < w\}$ and $\boldsymbol{y} = \boldsymbol{z}' \setminus \boldsymbol{x}$; then $(\, p_{\boldsymbol{az}})\varphi \triangleq (\, p_{\boldsymbol{a}'\boldsymbol{z}'})\, (\boldsymbol{x}, \overline{\boldsymbol{y}} < b\, w)\varphi$.

Now we can define a formula, in the style of $(a\, z\rangle_X$ (see Subsection 16), that checks for the executability of a pomset instead of a single event. As above, let $p_{\boldsymbol{az}} = (\boldsymbol{az}, <)$ be a poset, where $\boldsymbol{az} = a_1 z_1 \dots a_n z_n$ is a tuple of labelled variables. Then, for X a (finite) set of variables, we define

$$(\, p_{\boldsymbol{az}}\rangle_X\, \varphi \triangleq (\!(\, p_{\boldsymbol{az}})\!)\; (\langle X\rangle\, \langle \boldsymbol{z}\rangle\, \mathsf{T} \wedge \varphi)$$

Using this operator, we can extend the syntax of formulae given in Definition 23, allowing for checking the executability of a pomset, and so obtaining the logic for weak pomset h-bisimilarity.

Definition 26 *(logic for weak pomset h-bisimilarity).* Define, for a given tuple of labelled variables $\boldsymbol{ax}$

$$\varphi_{\boldsymbol{ax}} ::= \mathsf{T} \mid \neg\, \varphi_{\boldsymbol{ax}} \mid \varphi_{\boldsymbol{ax}} \wedge \varphi_{\boldsymbol{ax}} \mid (\, p_{\boldsymbol{bz}}\rangle_{\boldsymbol{x}}\, \varphi_{\boldsymbol{ax},\boldsymbol{bz}} \mid \bigvee_{\substack{\sigma : \boldsymbol{ax} \to \boldsymbol{ax} \\ \text{automorphism}}} \xi_{\boldsymbol{x}}\sigma$$

Then, $\mathcal{L}_{wph}$ is the set of formulae arising with $\boldsymbol{ax}$ empty, i.e., φ_ϵ.

We finally have:

Theorem 4 *(logical characterisation of $\approx_{\mathrm{wphb}}$). Let $\mathcal{E}_1$ and $\mathcal{E}_2$ be two image-finite PESs. Then, $\mathcal{E}_1 \equiv_{\mathcal{L}_{wph}} \mathcal{E}_2$ if and only if $\mathcal{E}_1 \approx_{\mathrm{wphb}} \mathcal{E}_2$.*

The result follows from the same intuitive reasoning described for weak h-bisimilarity, with the addition of pomsets. The presence of the pomset executability check operator $(\, p_{\boldsymbol{az}}\rangle_X\, \varphi$ ensures that the induced behavioural equivalence is (at least) a pomset bisimulation. Then, again, the other part of the syntax, using the execution-only formulae, guarantees that only isomorphic configurations can be related. In this way we have a pomset bisimulation relating only isomorphic configurations, that is, a weak pomset h-bisimulation. Formally, the proof is a routine adaptation of that of Theorem 3. The only crucial observation is that, given a poset $p_{\boldsymbol{az}} = (\boldsymbol{az}, <)$, whenever $\mathcal{E}, \varnothing \models_\eta (\, p_{\boldsymbol{az}}\rangle_X\, \varphi$, for an $\boldsymbol{ax}$-compatible environment η, then $\eta(X)$ is a configuration that can perform $\mathit{pomset}(p_{\boldsymbol{az}})$. In fact, the quantification part $(\!(\, p_{\boldsymbol{az}})\!)$ of the formula ensures that there exist events with the labels and causal dependencies corresponding to the pomset. Then, the satisfaction of $\langle X\rangle\, \langle \boldsymbol{z}\rangle\, \mathsf{T}$ ensures that the events bound to $\boldsymbol{z}$ are executable after $\eta(X)$, and thus no cause can be missing.

3.4. Distinguishing non-equivalent PESs, logically

We conclude this section by providing logical formulae that distinguish the PESs given in Figs. 2–10 with respect to the therein considered behavioural equivalences. The formulae provide a concise and formal description of the reason why every pair of PESs are not equivalent, according to the equivalence considered; hence, they formalise the intuitive arguments described in words in Examples 1–9.

Fig. 2: $\mathcal{E}_1 \not\approx_{\mathrm{st}} \mathcal{E}_2$

$$(\langle\!| a\, x |\!\rangle \otimes \langle\!| b\, y |\!\rangle)\mathsf{T} \quad \in \quad \mathcal{L}_s^t$$

This formula expresses the possibility of performing the step $\{\!|a\ b|\!\}$ and it is satisfied by $\mathcal{E}_1$ but not by $\mathcal{E}_2$.

Fig. 3: $\mathcal{E}_3 \not\approx_{\mathrm{ib}} \mathcal{E}_4$

$$\langle\!| a\, x |\!\rangle\, \neg \langle\!| b\, y |\!\rangle \mathsf{T} \quad \in \quad \mathcal{L}_i$$

This formula expresses the possibility of executing an a not followed by any b and it is satisfied by $\mathcal{E}_4$ but not by $\mathcal{E}_3$.

Fig. 4: $\mathcal{E}_5 \not\approx_{\mathrm{pt}} \mathcal{E}_6$

$$\langle\!| a\,x |\!\rangle\, \langle\!| x < b\,y |\!\rangle \mathsf{T} \quad \in \quad \mathcal{L}_p^t$$

This formula expresses the possibility of performing a pomset $\overset{b}{\underset{a}{\uparrow}}$ and it is satisfied by $\mathcal{E}_6$ but not by $\mathcal{E}_5$.

Fig. 5: $\mathcal{E}_7 \not\approx_{\mathrm{ib}} \mathcal{E}_8$

$$\langle\!| a\,x |\!\rangle (\langle\!| b\,y |\!\rangle \mathsf{T} \wedge \langle\!| c\,z |\!\rangle \mathsf{T}) \quad \in \quad \mathcal{L}_i$$

This formula expresses the possibility of choosing to perform b or c after the starting a and it is satisfied by $\mathcal{E}_7$ but not by $\mathcal{E}_8$.

Fig. 6: $\mathcal{E}_9 \not\approx_{\mathrm{pb}} \mathcal{E}_{10}$

$$\langle\!| a\,x |\!\rangle \neg \langle\!| a\,y |\!\rangle\, \langle\!| y < b\,z |\!\rangle \mathsf{T} \quad \in \quad \mathcal{L}_p$$

This formula expresses the possibility of performing an a after which the pomset $\overset{b}{\underset{a}{\uparrow}}$ cannot be executed any longer; it is satisfied by $\mathcal{E}_{10}$ but not by $\mathcal{E}_9$.

Fig. 7: $\mathcal{E}_{11} \not\approx_{\mathrm{whb}} \mathcal{E}_{12}$

$$(a\,x\rangle_{\emptyset}\, \neg(b\,y\rangle_{\{x\}}\, \langle y\rangle\, \langle x\rangle\, \mathsf{T} \quad \in \quad \mathcal{L}_{wh}$$

This formula says that there is an executable a for which there is no b, executable after the a, such that they can also be executed in the reverse order, hence they are causally independent; it is satisfied by $\mathcal{E}_{12}$ but not by $\mathcal{E}_{11}$.

Fig. 8: $\mathcal{E}_{13} \not\approx_{\mathrm{hb}} \mathcal{E}_{14}$

$$\langle\!| a\,x |\!\rangle\, \langle\!| \overline{x} < a\,y |\!\rangle\, (\langle\!| x < b\,z |\!\rangle \mathsf{T} \wedge \langle\!| y < b\,w |\!\rangle \mathsf{T}) \quad \in \quad \mathcal{L}_h$$

This formula says that there are two concurrent a's, each one causing some b immediately executable after them; this happens in $\mathcal{E}_{13}$ but not in $\mathcal{E}_{14}$.

Fig. 9: $\mathcal{E}_{15} \not\approx_{\mathrm{hhb}} \mathcal{E}_{16}$

$$(a\,x)\,(\overline{x} < b\,y)\,((x < c\,u)\mathsf{T} \wedge (y < c\,v)\mathsf{T} \wedge \neg(x, y < c\,z)\mathsf{T}) \quad \in \quad \mathcal{L}_{hh}$$

This formula says that there are two concurrent events, labelled by a and b, respectively, such that: (1) there is an event labelled c in the future caused by a; (2) the same holds for b as well; (3) there is no c in the future caused by both a and b. This formula is satisfied by $\mathcal{E}_{16}$: we can choose as the starting pair either a_2 and b_1, or a_3 and b_2. By contrast, the formula is not satisfied by $\mathcal{E}_{15}$: if we choose a_1 and b_1, condition (1) fails; if we choose a_2 and b_2, condition (2) fails; if we choose a_2 and b_1, condition (3) fails; if we choose a_1 and b_2, both conditions (1) and (2) fail.

Notice that the argument in Example 8 is much more involved than exhibiting this formula and arguing why $\mathcal{E}_{16}$ satisfies it, whereas $\mathcal{E}_{15}$ does not.

Fig. 10: $\mathcal{E}_{17} \ncong \mathcal{E}_{18}$

$$(a\,x)\,(a\,y)\,\neg\langle x\rangle\, \langle y\rangle\, \mathsf{T} \quad \in \quad \mathcal{L}$$

This formula states that there are two a's, possibly not distinct, such that either the first a cannot be executed or the second one cannot be executed after the first one. This happens in $\mathcal{E}_{18}$, where we can bind x and y to the two conflicting a's, but not in $\mathcal{E}_{17}$, where we are forced to bind x and y to the unique a, which is executable. Notice that the quantification of y in the formula is not well-formed, since the relation of y with x is not declared.

4. The spectrum for subclasses of PESs

In this section we study how the spectrum of behavioural equivalences changes when restricting the models to special subclasses of PESs, where either causality or conflicts are removed. Differently from [26,27], here we base our results on the logics characterising the various equivalences, identified in the previous section. The general pattern for showing that some equivalence is included into another will consist in proving that the logic for the latter equivalence can be encoded in the one for the former. Non-inclusion between equivalences will be witnessed by using distinguishing formulae, i.e., by providing structures which are identified by an equivalence and distinguished by a formula in the logics of the other equivalence.

$$\begin{tikzcd} \approx_{\mathrm{it}} = \approx_{\mathrm{st}} = \approx_{\mathrm{pt}} \\ \approx_{\mathrm{ib}} = \approx_{\mathrm{sb}} = \approx_{\mathrm{pb}} = \approx_{\mathrm{whb}} = \approx_{\mathrm{wphb}} = \approx_{\mathrm{hb}} \arrow[u] \\ \approx_{\mathrm{hhb}} \arrow[u] \\ \cong \arrow[u] \end{tikzcd}$$

Fig. 11. The spectrum for CSs.

4.1. Conflict without causality: coherence spaces

In this subsection we focus on the subclass of PESs obtained by considering an empty causality relation. These structures are also called *coherence spaces* [25], a model largely studied, e.g., in the field of linear logic and in the semantics of typed lambda-calculus [9,10,25].

Definition 27 *(coherence spaces).* A *coherence space* (written CS) is a PES $\mathcal{E}$ where the causality relation is empty.

In the setting of CSs, several definitions are radically simplified. For example, two events are concurrent if they are consistent, implying that any finite and consistent subset of E is a configuration. Moreover, a pomset in a CS is simply a multiset and, hence, pomsets and steps coincide.

Consequently, several behavioural equivalences collapse and the spectrum in Fig. 1 reduces to a chain. All trace equivalences coincide. They are properly coarser than bisimilarities, that all coincide, except for $\approx_{\mathrm{hhb}}$ that properly refines them. The latter, in turn, is still strictly coarser than isomorphism. The resulting spectrum for CSs is depicted in Fig. 11. While these results are already known from [26,27], here we show how they can be recovered for image-finite CSs by relying on the logical characterisation of behavioural equivalences.

First, we show that interleaving bisimilarity coincides with h-bisimilarity. This immediately follows from the fact that every closed formula φ of $\mathcal{L}_h$ can be encoded as a (closed) formula $enc_i(\varphi)$ of $\mathcal{L}_i$ such that $\mathcal{E} \models \varphi$ if and only if $\mathcal{E} \models enc_i(\varphi)$, for every CS $\mathcal{E}$.

Definition 28 *(encoding $\mathcal{L}_h$ into $\mathcal{L}_i$ for CSs).* The encoding procedure $enc_i(\cdot) : \mathcal{L}_h \to \mathcal{L}_i$ is inductively defined as follows:

$$\begin{aligned}
enc_i(\mathsf{T}) &= \mathsf{T} \\
enc_i(\neg\varphi) &= \neg enc_i(\varphi) \\
enc_i(\varphi_1 \wedge \varphi_2) &= enc_i(\varphi_1) \wedge enc_i(\varphi_2) \\
enc_i(\langle\!| \boldsymbol{x}, \overline{\boldsymbol{y}} < a\, z |\!\rangle\, \varphi) &= \begin{cases} \langle\!| az |\!\rangle\, enc_i(\varphi) & \text{if } \boldsymbol{x} = \epsilon \\ \mathsf{F} & \text{otherwise} \end{cases}
\end{aligned}$$

It is easy to see that the definition above is well-given, i.e., $enc_i(\varphi)$ is a formula of $\mathcal{L}_i$, for every formula φ of $\mathcal{L}_h$. The idea underlying the encoding is simple. The only subtle clause is the last one. Since causality is empty, every formula asking for the presence of causalities is false. Instead, since all events that can be executed in the same computation are concurrent, the requirement of concurrency is voided.

Proposition 2 *(soundness of the encoding). Let $\mathcal{E}$ be a CS and let φ be a closed formula of $\mathcal{L}_h$. Then, $\mathcal{E} \models \varphi$ if and only if $\mathcal{E} \models enc_i(\varphi)$.*

Proof. We show this by proving a more general property, that is, for every formula φ of $\mathcal{L}_h$, configuration $C \in Conf(\mathcal{E})$ and environment $\eta \in Env_{\mathcal{E}}$ such that $\eta(fv(\varphi)) \subseteq C$, it holds that $\mathcal{E}, C \models_\eta \varphi$ if and only if $\mathcal{E}, C \models_\eta enc_i(\varphi)$. This clearly implies the desired result for $C = \emptyset$ when φ is closed. We proceed by induction on the formula φ.

- $\varphi = \mathsf{T}$ and hence $enc_i(\varphi) = \mathsf{T}$: Immediate.
- $\varphi = \neg\psi$ and hence $enc_i(\varphi) = \neg enc_i(\psi)$: By definition of the semantics, $\mathcal{E}, C \models_\eta \neg\psi$ iff $\mathcal{E}, C \not\models_\eta \psi$, which, by inductive hypothesis, happens iff $\mathcal{E}, C \not\models_\eta enc_i(\psi)$, which in turn, by definition of the semantics, holds iff $\mathcal{E}, C \models_\eta \neg enc_i(\psi)$.
- $\varphi = \psi_1 \wedge \psi_2$ and hence $enc_i(\varphi) = enc_i(\psi_1) \wedge enc_i(\psi_2)$: By definition of the semantics, $\mathcal{E}, C \models_\eta \psi_1 \wedge \psi_2$ iff $\mathcal{E}, C \models_\eta \psi_1$ and $\mathcal{E}, C \models_\eta \psi_2$, which, by inductive hypothesis, happen iff $\mathcal{E}, C \models_\eta enc_i(\psi_1)$ and $\mathcal{E}, C \models_\eta enc_i(\psi_2)$. Again by definition of the semantics, this holds iff $\mathcal{E}, C \models_\eta enc_i(\psi_1) \wedge enc_i(\psi_2)$.

- $\varphi = \langle\!| \overline{\boldsymbol{y}} < az |\!\rangle\, \psi$ and hence $enc_i(\varphi) = \langle\!| az |\!\rangle\, enc_i(\psi)$: By definition of the semantics $\mathcal{E}, C \models_\eta \langle\!| \overline{\boldsymbol{y}} < az |\!\rangle\, \psi$ iff there is a transition $C \xrightarrow{l(e)} C' = C \cup \{e\}$ s.t. $e \parallel \eta(\boldsymbol{y})$ and $\mathcal{E}, C' \models_{\eta[z\mapsto e]} \psi$. Observe that the requirement $e \parallel \eta(\boldsymbol{y})$ is irrelevant since CSs have empty causality relation. Since $fv(\psi) \subseteq fv(\varphi) \cup \{z\}$, $\eta[z \mapsto e](z) \in C'$ and by hypothesis $\eta(\boldsymbol{x}) \subseteq C \subset C'$, we must have that $\eta[z \mapsto e](fv(\psi)) \subseteq C'$. Then, by inductive hypothesis, we deduce that $\mathcal{E}, C \models_\eta \langle\!| \overline{\boldsymbol{y}} < az |\!\rangle\, \psi$ iff there is a transition $C \xrightarrow{l(e)} C'$ such that $\mathcal{E}, C' \models_{\eta[z\mapsto e]} enc_i(\psi)$, which, by definition of the semantics, holds iff $\mathcal{E}, C \models_\eta \langle\!| az |\!\rangle\, enc_i(\psi)$.
- $\varphi = \langle\!| \boldsymbol{x}, \overline{\boldsymbol{y}} < a\, z |\!\rangle\, \psi$, for $\boldsymbol{x} \neq \epsilon$, and hence $enc_i(\varphi) = \mathsf{F}$: By definition of the semantics, it always holds that $\mathcal{E}, C \not\models_\eta \mathsf{F}$. So we just need to prove that also $\mathcal{E}, C \not\models_\eta \langle\!| \boldsymbol{x}, \overline{\boldsymbol{y}} < a\, z |\!\rangle\, \psi$. But, clearly, this holds as well, since by hypothesis $\eta(\boldsymbol{x}) \subseteq C$ and CSs have empty causality relation, therefore the diamond operator cannot be ever satisfied. □

Corollary 1 *($\approx_{\mathrm{hb}}$ collapses to $\approx_{\mathrm{ib}}$ for CSs). Let $\mathcal{E}$ and $\mathcal{F}$ be CSs. If $\mathcal{E} \approx_{\mathrm{ib}} \mathcal{F}$ then $\mathcal{E} \approx_{\mathrm{hb}} \mathcal{F}$*

Proof. Immediate consequence of Theorem 1 and Proposition 2. □

Interestingly, we can exploit Proposition 2 also to show that, on CSs, interleaving trace equivalence coincides with pomset trace equivalence. In fact, it can be seen that, for every formula φ of $\mathcal{L}_p^t$, the formula $enc_i(\varphi)$ is actually a formula of $\mathcal{L}_i^t$. Then, by Proposition 2 (and Theorem 1) we immediately obtain the desired result.

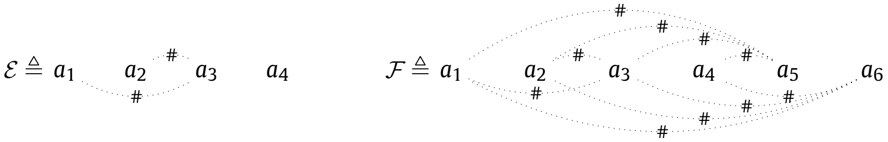
Corollary 2 *($\approx_{\mathrm{pt}}$ collapses to $\approx_{\mathrm{it}}$ for CSs). Let $\mathcal{E}$ and $\mathcal{F}$ be CSs. If $\mathcal{E} \approx_{\mathrm{it}} \mathcal{F}$ then $\mathcal{E} \approx_{\mathrm{pt}} \mathcal{F}$.*

In the rest of this subsection, we exhibit separating formulae for those equivalences that remain different.

- $\approx_{\mathrm{ib}} \subsetneq \approx_{\mathrm{it}}$: Consider the CSs from [26, Prop. 3]:

  $$\mathcal{E} \triangleq a_1 \cdots\#\cdots a_3 \cdots\#\cdots a_2 \qquad\qquad \mathcal{F} \triangleq a_1 \quad a_2$$

  They are interleaving trace equivalent, i.e., $\mathcal{E} \approx_{\mathrm{it}} \mathcal{F}$. In fact, $SeqTr(\mathcal{E}) = SeqTr(\mathcal{F}) = \{\epsilon, a, aa\}$. The fact that $\mathcal{E} \not\approx_{\mathrm{ib}} \mathcal{F}$ can be witnessed by a formula in $\mathcal{L}_i$, namely $\varphi = \langle\!| a\, x |\!\rangle \neg \langle\!| a\, y |\!\rangle \mathsf{T}$, stating that there exists an a after which no other a is available. In fact, it is easily seen that $\mathcal{E} \models \varphi$ while $\mathcal{F} \not\models \varphi$. Note that φ essentially relies on negation, hence it is not part of $\mathcal{L}_i^t$.
- $\approx_{\mathrm{hhb}} \subsetneq \approx_{\mathrm{hb}}$: Consider the CSs from [26, Prop. 4]:

  It can be seen that $\mathcal{E} \approx_{\mathrm{hb}} \mathcal{F}$ by providing a h-bisimulation. The bisimulation associates a_i in $\mathcal{E}$ with a_i in $\mathcal{F}$, for $i = 1, \ldots, 4$. The delicate issue is when $\mathcal{F}$ challenges with a_5 or a_6: in both cases, the first reply of $\mathcal{E}$ must be with a_3, to definitely exclude a_1 and a_2, and leave only a_4 enabled for execution; for more details, see [26].
  The fact that $\mathcal{E} \not\approx_{\mathrm{hhb}} \mathcal{F}$ is witnessed by a formula in $\mathcal{L}_{hh}$, e.g. $\varphi = ((a\, x) \otimes (a\, y))((\langle x \rangle \neg((a\, z) \otimes (a\, w))\mathsf{T}) \wedge (\langle y \rangle \neg((a\, z) \otimes (a\, w))\mathsf{T}))$. Intuitively, it states that there are two concurrent a's such that no other a's is concurrent with either of the two (and thus, since we are in CSs, all remaining as are in conflict with both of them). Clearly, $\mathcal{F} \models \varphi$ by binding x and y to the two rightmost a-events. By contrast, $\mathcal{E} \not\models \varphi$. In fact, in whichever way we bind x and y, there will always be at least one further a that is concurrent with either x or y (or both).
- $\cong \subsetneq \approx_{\mathrm{hhb}}$: Consider the CSs from [26, Prop. 5]:

  $$\mathcal{E} \triangleq a_1 \cdots\#\cdots a_2 \qquad\qquad \mathcal{F} \triangleq a$$

  It can be seen that $\mathcal{E} \approx_{\mathrm{hhb}} \mathcal{F}$. In fact, the relation $R = \{(\emptyset, \emptyset, \emptyset), (\{a_1\}, \{a\}, [a_1 \mapsto a]), (\{a_2\}, \{a\}, [a_2 \mapsto a])\}$ is a hereditary h-bisimulation. Clearly, $\mathcal{E} \not\cong \mathcal{F}$ and, indeed, in $\mathcal{L}$ we can find the formula $\varphi = (a\, x)(a\, y)\langle x \rangle \neg \langle y \rangle \mathsf{T}$ such that $\mathcal{E} \models \varphi$ while $\mathcal{F} \not\models \varphi$.

4.2. Causality without conflict: elementary ESs

A second restriction of PESs is obtained by considering an empty conflict relation; this yields *elementary event structures* [42].

Definition 29 *(elementary event structures).* An *elementary event structure* (written EES) is a PES $\mathcal{E}$ where the conflict relation is empty.

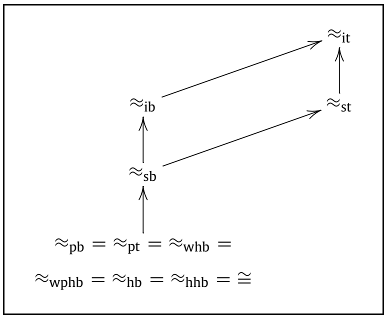

$\approx_{\mathrm{it}}$
$\approx_{\mathrm{ib}}$
$\approx_{\mathrm{st}}$
$\approx_{\mathrm{sb}}$
$\approx_{\mathrm{pb}} = \approx_{\mathrm{pt}} = \approx_{\mathrm{whb}} =$
$\approx_{\mathrm{wphb}} = \approx_{\mathrm{hb}} = \approx_{\mathrm{hhb}} = \cong$

Fig. 12. The spectrum for (image-finite) EESs.

Having an empty conflict relation, makes the picture simpler: two events are always concurrent when they are not causally dependent. The spectrum of Fig. 1 reduces to that in Fig. 12 where some equivalences collapse (less than what happens for CSs).

In this situation, pomset trace equivalence, and thus all the equivalences in between, collapses to isomorphism. This fact follows by observing that, whenever two EESs are logically equivalent under $\mathcal{L}_p^t$, then they are isomorphic.

Proposition 3 *($\equiv_{\mathcal{L}_p^t}$ is isomorphism for EESs). Let $\mathcal{E}_1$ and $\mathcal{E}_2$ be two image-finite EESs. If $\mathcal{E}_1 \equiv_{\mathcal{L}_p^t} \mathcal{E}_2$ then $\mathcal{E}_1 \cong \mathcal{E}_2$.*

Proof. Assume that $\mathcal{E}_1 \equiv_{\mathcal{L}_p^t} \mathcal{E}_2$. Then, for every closed formula φ of $\mathcal{L}_p^t$, we know that $\mathcal{E}_1$ satisfies φ if and only if $\mathcal{E}_2$ does. Observe that, since $\mathcal{E}_1$ and $\mathcal{E}_2$ are image-finite EESs, for each $k \in \mathbb{N}$, the corresponding k-prefixes $\mathcal{E}_1^{(k)}$ and $\mathcal{E}_2^{(k)}$ are both finite pomsets, by Lemma 1 and the fact that EESs have empty conflict relation. Therefore, there must be a closed formula φ_1 of $\mathcal{L}_p^t$ requiring exactly the possibility to execute the pomset $\mathcal{E}_1^{(k)}$, and, similarly, a closed formula φ_2 for the pomset $\mathcal{E}_2^{(k)}$. Since we know that $\mathcal{E}_1$ and $\mathcal{E}_2$ satisfy the same closed formulae of $\mathcal{L}_p^t$, and each formula is surely satisfied by the corresponding EES, we have that both EESs satisfy both formulae. Thus, the two pomsets must be isomorphic. Then, we can immediately conclude that $\mathcal{E}_1 \cong \mathcal{E}_2$ by Lemma 2. □

For interleaving and step equivalences, instead, the spectrum for EESs is the same as that for general PESs: the (strict) inclusions depicted in the upper part of Fig. 1 also hold for EESs. Here we provide examples of EESs distinguished by formulae of the proper fragments showing these facts.

- $\approx_{\mathrm{ib}} \not\subseteq \approx_{\mathrm{st}}$: Consider the EESs from [26, Prop. 6]:

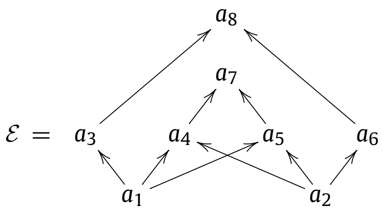
$$\begin{tikzcd} \mathcal{E} \triangleq a_2 & & \mathcal{F} \triangleq a_1 & a_2 \\ a_1 \arrow[u] & & & \end{tikzcd}$$

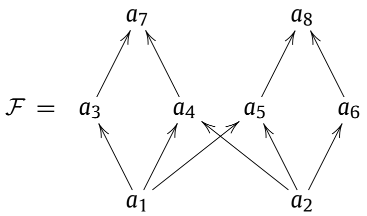
  They are trivially interleaving equivalent (both trace and bisimulation), whereas they are not step-trace equivalent. This can be witnessed by a formula in $\mathcal{L}_s^t$, namely $(\langle\!| a\, x |\!\rangle \otimes \langle\!| a\, y |\!\rangle)\mathrm{T}$, satisfied by $\mathcal{F}$ and not by $\mathcal{E}$.
- $\approx_{\mathrm{st}} \not\subseteq \approx_{\mathrm{ib}}$: Consider the EESs from [26, Prop. 8]:

$$\begin{tikzcd} \mathcal{E} = & b_1 & b_2 & & \mathcal{F} = & b_1 & b_2 \\ & a_1 \arrow[u] \arrow[ur] & a_2 \arrow[u] & & & a_1 \arrow[u] & a_2 \arrow[u] \end{tikzcd}$$

  We have that $\mathcal{E} \approx_{\mathrm{st}} \mathcal{F}$, in fact $\mathit{StepTr}(\mathcal{E}) = \mathit{StepTr}(\mathcal{F}) = \{\epsilon, a, aa, ab, aab, aba, aabb, abab, \{\!\{aa\}\!\}, a\{\!\{ab\}\!\}, aa\{\!\{bb\}\!\}, a\{\!\{ab\}\!\}b, \{\!\{aa\}\!\}bb, \{\!\{aa\}\!\}\{\!\{bb\}\!\}\}$. Instead, $\mathcal{E} \not\approx_{\mathrm{ib}} \mathcal{F}$, as it can be witnessed by a formula in $\mathcal{L}_i$, namely $\langle\!| a\, x |\!\rangle \neg \langle\!| b\, y |\!\rangle \mathrm{T}$, satisfied by $\mathcal{E}$ and not by $\mathcal{F}$.
- As corollary of the two results above, we obtain that, for EESs, $\approx_{\mathrm{ib}}$ and $\approx_{\mathrm{st}}$ do not contain $\approx_{\mathrm{it}}$, nor either of them is contained in $\approx_{\mathrm{sb}}$. This can be easily deduced from the incomparability of the two equivalences proved above. For instance, if $\approx_{\mathrm{ib}}$ would contain (hence collapse on) $\approx_{\mathrm{it}}$, since we already know that for general PESs $\approx_{\mathrm{it}}$ includes $\approx_{\mathrm{st}}$, then we would obtain that $\approx_{\mathrm{ib}}$ includes $\approx_{\mathrm{st}}$.

- $\approx_{\mathrm{sb}} \not\subseteq \approx_{\mathrm{pt}}$: Consider the EESs from [26, Prop. 9]:





It can be seen that $\mathcal{E} \approx_{\mathrm{sb}} \mathcal{F}$ (see [26,27]). The fact that $\mathcal{E} \not\approx_{\mathrm{pt}} \mathcal{F}$ is witnessed by a formula in $\mathcal{L}^t_p$, i.e., $\langle\!| a\,x_0 |\!\rangle \langle\!| a\,x_1 |\!\rangle \langle\!| x_0, x_1 < a\,x_3 |\!\rangle \langle\!| x_0, x_1 < a\,x_4 |\!\rangle \langle\!| x_3, x_4 < a\,x_6 |\!\rangle \mathsf{T}$, satisfied by $\mathcal{E}$ and not by $\mathcal{F}$.

- An easy corollary of the previous result is that, for EESs, $\approx_{\mathrm{sb}}$ is not contained in $\approx_{\mathrm{pb}}$, $\approx_{\mathrm{whb}}$, $\approx_{\mathrm{wphb}}$, $\approx_{\mathrm{hb}}$, $\approx_{\mathrm{hhb}}$, and $\cong$.

5. Conclusion

We proposed an enhancement of the logical framework developed in [3] able to fully characterise the whole spectrum of behavioural equivalences provided in [21,51]. In particular, we showed that, by removing a well-formedness condition from the logic introduced in [3] for characterising hereditary history preserving bisimilarity, the logic becomes expressive enough to tell apart non-isomorphic PESs. We then distilled sublogics able to characterise all the equivalences in the true concurrent spectrum for image-finite PESs. We applied the logic framework developed to rediscover the spectra of equivalences provided in [26,27] for PESs where either causality or conflict is removed, namely in the framework of coherence spaces and elementary event structures, respectively.

A natural question regards the possibility of obtaining analogous results for event structure models beyond PESs. The extension to models where some notion of causality can be defined, at least locally to each configuration (like in stable event structures [55]), could require little modifications to the logics. Instead, for unstable models, one could still explore the potentiality of a variant of the logics $\mathcal{L}$ investigated in [2], where causality and concurrency are not used explicitly. Indeed, in unstable models, causality becomes ambiguous also within configurations, because an event may be enabled in more than one way even in a single configuration. Still, in each configuration every event has a well-defined set of possible causes, as illustrated by Winskel's example of a parallel switch [55, p. 328]. In a companion paper [4], we provide a first attempt in this direction and study history preserving behavioural equivalences over *configuration structures* [53], a model which generalises the families of (finite) configurations of event structures. There, we show that history preserving bisimilarity and its hereditary version admit a particularly simple characterisation, similar in spirit to those for prime and stable event structures. This is exploited for identifying two behavioural logics that characterise such equivalences.

In the same direction, one could try to investigate classes of models with a more involved interplay of causality and conflict. An interesting example is basic parallel processes (BPPs) [14]. Recently many results have been attained for subclasses of these models such as simple (S)BPPs [24], i.e., BPPs where summation and recursion are required to be guarded, and the corresponding BPP nets [28]. It would be worthwhile to study the spectrum of equivalences in the setting of BPPs and SBPPs, which identify subclasses of PESs, clearly more expressive than CSs and EESs. Such a goal is beyond the scope of this paper, but the mentioned results in the literature and some preliminary investigation show that, in fact, in such setting the behavioural equivalences in the spectrum have less coincidences than on CSs and EESs. For instance, pomset bisimilarity and weak h-bisimilarity, which coincide in the case of both CSs and EESs, turn out to be distinct in the case of BPPs. Indeed, consider the BPPs $P = a.(b+c) + (a \parallel b)$ and $Q = a.(b+c) + (a \parallel b) + a.b$, borrowed from [11] (where they are shown to be pomset bisimilar): they are distinguished by the $\mathcal{L}_{wh}$ formula $(a\,x\rangle_{\varnothing}((b\,y\rangle_{\{x\}}\neg\langle y\rangle\,\langle x\rangle\,\mathsf{T} \wedge \neg(c\,z\rangle_{\{x\}}\mathsf{T})$, requiring an a causing b but not allowing the execution of c: the formula holds only on Q, and hence P and Q are not weak h-bisimilar. However, the same might not be true for the subclass of SBPPs. Indeed, not all summations in the processes above are guarded, and it does not seem possible to make them so while preserving the (in)equivalence results.

In recent works [29,30], a novel bisimulation, named *team bisimilarity*, was introduced and proved to be finer than interleaving bisimilarity; actually, this coincides with a slight strengthening of h-bisimilarity on BPP nets. Furthermore, a logic, called *team modal logic* (TML) and that induces such an equivalence, was also introduced. Like our logic $\mathcal{L}$, also TML is an extension of Hennessy-Milner logic [34]. Given the similarities, an interesting research direction would be to identify, if possible, a fragment of our logic $\mathcal{L}$ inducing team bisimilarity on BPP nets, and to confront it with TML. Additionally, one could try a logical investigation of other equivalences (e.g., those presented in [52, Sect. 3] or other hereditary equivalences as those in [46]). In particular, although this is not explored in full detail in this paper, we believe that the derived operator $(a\,z\rangle_X$, in combination with execution-only formulae, could be used to uniformly characterise all equivalences up to history preserving bisimilarity: for basic equivalences (interleaving, step, pomset bisimilarity), execution-only formulae would be omitted; for the corresponding weak history preserving equivalences, execution-only formulae are used in a constrained way as we did in this paper; for history preserving equivalences, they are freely used.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We are grateful to the anonymous reviewers for their careful reading and for their suggestions that improved our work.

Appendix A. Omitted proofs from Section 3

Proof of Lemma 6. (1)⇒(2). Assume $\mathit{poset}(\eta_1(\boldsymbol{x})) \cong \mathit{poset}(\eta_2(\boldsymbol{x}))$, hence there is an isomorphism $f : \eta_1(\boldsymbol{x}) \to \eta_2(\boldsymbol{x})$. Let $\varphi_{\boldsymbol{ax}} = \bigvee_{\substack{\sigma:\boldsymbol{ax}\to\boldsymbol{ax}\\ \text{automorphism}}} \xi_{\boldsymbol{x}}\sigma$. We just show that $\mathcal{E}_1, \varnothing \models_{\eta_1} \varphi_{\boldsymbol{ax}}$ implies $\mathcal{E}_2, \varnothing \models_{\eta_2} \varphi_{\boldsymbol{ax}}$ since the other direction is symmetric. From the premise, by definition of the semantics, we know that there exists an automorphism $\sigma_1 : \boldsymbol{ax} \to \boldsymbol{ax}$ such that $\mathcal{E}_1, \varnothing \models_{\eta_1} \xi_{\boldsymbol{x}}\sigma_1$. Let $\sigma_2 : \boldsymbol{ax} \to \boldsymbol{ax}$ be defined as $\sigma_2(x) = y$ such that $\eta_2(y) = f(\eta_1(\sigma_1(x)))$. Since η_1 and η_2 are $\boldsymbol{ax}$-compatible, f is an isomorphism and σ_1 preserves the labels, we have that σ_2 is a bijection which preserves the labels, hence an automorphism over $\boldsymbol{ax}$. Then, it is enough to prove that $\mathcal{E}_2, \varnothing \models_{\eta_2} \xi_{\boldsymbol{x}}\sigma_2$, which implies $\mathcal{E}_2, \varnothing \models_{\eta_2} \varphi_{\boldsymbol{ax}}$.

We actually prove a stronger property, that is, given an execution-only formula ξ such that $\mathit{fv}(\xi) \subseteq \boldsymbol{x}$, and two configurations $C_1 \in \mathit{Conf}(\mathcal{E}_1)$ and $C_2 \in \mathit{Conf}(\mathcal{E}_2)$ such that $C_1 \subseteq \eta_1(\boldsymbol{x})$ and $C_2 = f(C_1) \subseteq \eta_2(\boldsymbol{x})$, it holds that $\mathcal{E}_1, C_1 \models_{\eta_1} \xi\sigma_1$ if and only if $\mathcal{E}_2, C_2 \models_{\eta_2} \xi\sigma_2$. This clearly encompasses the more specific case above where $C_1 = C_2 = \varnothing$. We proceed by induction on the shape of ξ. We discuss only some cases and a single direction, the other being symmetric.

$\xi = \neg\xi'$: Assume that $\mathcal{E}_1, C_1 \models_{\eta_1} \neg\xi'\sigma_1$. By definition of the semantics, we know that $\mathcal{E}_1, C_1 \not\models_{\eta_1} \xi'\sigma_1$. Then, since $\mathit{fv}(\xi') = \mathit{fv}(\xi) \subseteq \boldsymbol{x}$, by inductive hypothesis we have that $\mathcal{E}_2, C_2 \not\models_{\eta_2} \xi'\sigma_2$. Again by definition of the semantics, we conclude that $\mathcal{E}_2, C_2 \models_{\eta_2} \neg\xi'\sigma_2$.

$\xi = \langle z\rangle\, \xi'$: Assume that $\mathcal{E}_1, C_1 \models_{\eta_1} \langle \sigma_1(z)\rangle\, (\xi'\sigma_1)$. By definition of the semantics, we know that either the event $e = \eta_1(\sigma_1(z)) \in C_1$ and $\mathcal{E}_1, C_1 \models_{\eta_1} \xi'\sigma_1$, or $e = \eta_1(\sigma_1(z))$ can be executed from C_1 obtaining a configuration $C_1' = C_1 \cup \{e\}$ such that $\mathcal{E}_1, C_1' \models_{\eta_1} \xi'\sigma_1$.

In the first case, since $C_2 = f(C_1)$, we must have that $\eta_2(\sigma_2(z)) = f(\eta_1(\sigma_1(z))) = f(e) \in C_2$. Furthermore, since $\mathit{fv}(\xi') \subseteq \mathit{fv}(\xi) \subseteq \boldsymbol{x}$, by inductive hypothesis we have that $\mathcal{E}_2, C_2 \models_{\eta_2} \xi'\sigma_2$.

In the second case, observe that e must have been minimal in the poset $\eta_1(\boldsymbol{x}) \setminus C_1$, and the same must hold for $\eta_2(\sigma_2(z)) = f(e)$ in $\eta_2(\boldsymbol{x}) \setminus C_2$, since $C_2 = f(C_1)$. Therefore, $f(e)$ can be executed from C_2 obtaining the configuration $C_2' = C_2 \cup \{f(e)\}$. Since $\mathit{fv}(\xi') \subseteq \mathit{fv}(\xi) \subseteq \boldsymbol{x}$, $C_1' \subseteq \eta_1(\boldsymbol{x})$, $C_2' \subseteq \eta_2(\boldsymbol{x})$ and $C_2' = f(C_1')$, by inductive hypothesis we have that $\mathcal{E}_2, C_2' \models_{\eta_2} \xi'\sigma_2$.

Thus, in both cases, by the properties of $f(e)$ above, we can conclude that $\mathcal{E}_2, C_2 \models_{\eta_2} \langle \sigma_2(z)\rangle\, (\xi'\sigma_2)$.

(2)⇒(1). Assume that $\mathcal{E}_1, \varnothing \models_{\eta_1} \varphi_{\boldsymbol{ax}}$ if and only if $\mathcal{E}_2, \varnothing \models_{\eta_2} \varphi_{\boldsymbol{ax}}$ for every $\boldsymbol{ax}$-parameterised formula $\varphi_{\boldsymbol{ax}}$ of the form $\bigvee_{\substack{\sigma:\boldsymbol{ax}\to\boldsymbol{ax}\\ \text{automorphism}}} \xi_{\boldsymbol{x}}\sigma$. In particular, this holds for

$$\varphi_{\boldsymbol{ax}} = \bigvee_{\substack{\sigma:\boldsymbol{ax}\to\boldsymbol{ax}\\ \text{automorphism}}} \left(\bigwedge_{\substack{(\eta_1(x_1),\dots,\eta_1(x_n))\in \mathit{lin}(\eta_1(\boldsymbol{x}))\\ \{x_1,\dots,x_n\}=\boldsymbol{x}}} \langle\sigma(x_1)\rangle \dots \langle\sigma(x_n)\rangle\, \mathsf{T} \wedge \bigwedge_{\substack{(\eta_1(y_1),\dots,\eta_1(y_n))\notin \mathit{lin}(\eta_1(\boldsymbol{x}))\\ \{y_1,\dots,y_n\}=\boldsymbol{x}}} \neg\langle\sigma(y_1)\rangle \dots \langle\sigma(y_n)\rangle\, \mathsf{T} \right)$$

where $\mathit{lin}(\cdot)$ denotes all the linearisations of a set of events (that respects causes).

We first show that $\mathcal{E}_1, \varnothing \models_{\eta_1} \varphi_{\boldsymbol{ax}}$. Take $\sigma : \boldsymbol{ax} \to \boldsymbol{ax}$ as the identity function. Observe that, given a linearisation $(e_1, \dots, e_n) \in \mathit{lin}(E)$ of some E, it is guaranteed that e_i is minimal in $E \setminus \{e_1, \dots, e_{i-1}\}$. This means that, for every linearisation $(\eta_1(x_1), \dots, \eta_1(x_n))$ of $\eta_1(\boldsymbol{x})$, since η_1 is $\boldsymbol{ax}$-compatible, we immediately have that $\mathcal{E}_1, \varnothing \models_{\eta_1} \langle x_1\rangle \dots \langle x_n\rangle\, \mathsf{T}$. On the other hand, given a sequence $(\eta_1(y_1), \dots, \eta_1(y_n))$ which is not a linearisation of $\eta_1(\boldsymbol{x})$, let $i, j \in \{1, \dots, n\}$ be such that $i < j$ and $\eta_1(y_j) \leq_{E_1} \eta_1(y_i)$. From the configuration $\{\eta_1(y_1), \dots, \eta_1(y_{i-1})\}$, if even reachable, it will not be possible to execute $\eta_1(y_i)$ since it is caused by $\eta_1(y_j)$ which has not been executed yet. Therefore, we must have that $\mathcal{E}_1, \varnothing \not\models_{\eta_1} \langle y_1\rangle \dots \langle y_n\rangle\, \mathsf{T}$, hence $\mathcal{E}_1, \varnothing \models_{\eta_1} \neg\langle y_1\rangle \dots \langle y_n\rangle\, \mathsf{T}$.

Then, by hypothesis we have that also $\mathcal{E}_2, \varnothing \models_{\eta_2} \varphi_{\boldsymbol{ax}}$. By definition of the semantics, this means that there exists an automorphism σ over $\boldsymbol{ax}$ such that, for every linearisation $(\eta_1(x_1), \dots, \eta_1(x_n))$ of $\eta_1(\boldsymbol{x})$, it holds that $\mathcal{E}_2, \varnothing \models_{\eta_2} \langle\sigma(x_1)\rangle \dots \langle\sigma(x_n)\rangle\, \mathsf{T}$, and, for every sequence $(\eta_1(y_1), \dots, \eta_1(y_n))$ which is not a linearisation of $\eta_1(\boldsymbol{x})$, it holds that $\mathcal{E}_2, \varnothing \not\models_{\eta_2} \langle\sigma(y_1)\rangle \dots \langle\sigma(y_n)\rangle\, \mathsf{T}$. Observe that the function $f : \eta_1(\boldsymbol{x}) \to \eta_2(\boldsymbol{x})$ defined by $f = \eta_2 \circ \sigma \circ \eta_1^{-1}$ is a well-defined bijection which preserves the labels, because η_1 and η_2 are $\boldsymbol{ax}$-compatible. Moreover, the image through f of every linearisation $(\eta_1(x_1), \dots, \eta_1(x_n))$ of $\eta_1(\boldsymbol{x})$ is a linearisation of $\eta_2(\boldsymbol{x})$, since $f(\eta_1(x_i)) = \eta_2(\sigma(x_i))$ for all $i \in \{1, \dots, n\}$, and $\mathcal{E}_2, \varnothing \models_{\eta_2} \langle \sigma(x_1)\rangle \dots \langle \sigma(x_n)\rangle \mathsf{T}$. On the other hand, for every sequence $(\eta_1(y_1), \dots, \eta_1(y_n))$ which is not a linearisation of $\eta_1(\boldsymbol{x})$, its image through f is not a linearisation of $\eta_2(\boldsymbol{x})$, since $f(\eta_1(y_i)) = \eta_2(\sigma(y_i))$ for all $i \in \{1, \dots, n\}$, and $\mathcal{E}_2, \varnothing \not\models_{\eta_2} \langle \sigma(y_1)\rangle \dots \langle \sigma(y_n)\rangle \mathsf{T}$.

Therefore, we deduce that $lin(\eta_2(\boldsymbol{x})) = f(lin(\eta_1(\boldsymbol{x})))$, and so f is actually an isomorphism between $\eta_1(\boldsymbol{x})$ and $\eta_2(\boldsymbol{x})$. Indeed, for every pair of variables $x, x' \in \boldsymbol{x}$, we have $\eta_1(x) \leq_{E_1} \eta_1(x')$ if and only if $\eta_1(x)$ appears before $\eta_1(x')$ in every linearisation of $\eta_1(\boldsymbol{x})$, which holds if and only if $f(\eta_1(x))$ appears before $f(\eta_1(x'))$ in every linearisation of $\eta_2(\boldsymbol{x})$, since $lin(\eta_2(\boldsymbol{x})) = f(lin(\eta_1(\boldsymbol{x})))$, which holds if and only if $f(\eta_1(x)) \leq_{E_2} f(\eta_1(x'))$. □

Proof of Lemma 7. We proceed by induction on the shape of the formula $\varphi_{\boldsymbol{ax}}$. We discuss only some cases and a single direction, the other being symmetric.

$\varphi_{\boldsymbol{ax}} = \neg\, \psi_{\boldsymbol{ax}}$**:** Assume that $\mathcal{E}, \varnothing \models_\eta \neg\, \psi_{\boldsymbol{ax}}$. By definition of the semantics, we know that $\mathcal{E}, \varnothing \not\models_\eta \psi_{\boldsymbol{ax}}$. Then, since η and $\sigma \circ \eta$ are still $\boldsymbol{ax}$-compatible, by inductive hypothesis we have that $\mathcal{E}, \varnothing \not\models_{\sigma\circ\eta} \psi_{\boldsymbol{ax}}$. Again by definition of the semantics, we conclude that $\mathcal{E}, \varnothing \models_{\sigma\circ\eta} \neg\, \psi_{\boldsymbol{ax}}$.

$\varphi_{\boldsymbol{ax}} = (b\, z\rangle_{\boldsymbol{x}}\, \psi_{\boldsymbol{a'x'}}$**, where** $\boldsymbol{a'x'} = \boldsymbol{ax}, bz$**:** Assume that $\mathcal{E}, \varnothing \models_\eta (b\, z\rangle_{\boldsymbol{x}}\, \psi_{\boldsymbol{a'x'}}$. By definition of the semantics, we know that there exists an event $e \in E$ such that $l(e) = b$, $\varnothing \to \dots \to \eta(\boldsymbol{x}) \xrightarrow{b} \eta(\boldsymbol{x}) \cup \{e\} = \eta[z \mapsto e](\boldsymbol{x'})$ and $\mathcal{E}, \varnothing \models_{\eta[z\mapsto e]} \psi_{\boldsymbol{a'x'}}$. Observe that the function $\sigma' = \sigma \cup \{(e,e)\}$ is a bijection over $\eta[z \mapsto e](\boldsymbol{x'})$. Then, since $\eta[z \mapsto e]$ and $\sigma' \circ (\eta[z \mapsto e])$ are both $\boldsymbol{a'x'}$-compatible, by inductive hypothesis we have that $\mathcal{E}, \varnothing \models_{\sigma'\circ(\eta[z\mapsto e])} \psi_{\boldsymbol{a'x'}}$. Furthermore, since $\sigma' \circ (\eta[z \mapsto e]) = (\sigma \circ \eta)[z \mapsto e]$, we know that $\mathcal{E}, \varnothing \models_{(\sigma\circ\eta)[z\mapsto e]} \psi_{\boldsymbol{a'x'}}$. Then, for the same transition above $\sigma(\eta(\boldsymbol{x})) \xrightarrow{b} \sigma(\eta(\boldsymbol{x})) \cup \{e\} = (\sigma \circ \eta)[z \mapsto e](\boldsymbol{x'})$, we can conclude that $\mathcal{E}, \varnothing \models_{\sigma\circ\eta} (b\, z\rangle_{\boldsymbol{x}}\, \psi_{\boldsymbol{a'x'}}$.

$\varphi_{\boldsymbol{ax}} = \bigvee_{\substack{\sigma' : \boldsymbol{ax}\to\boldsymbol{ax} \\ \textbf{automorphism}}} \xi_{\boldsymbol{x}}\sigma'$**:** Immediate by Lemma 6, since η and $\sigma \circ \eta$ are $\boldsymbol{ax}$-compatible, and $\eta(\boldsymbol{x}) = (\sigma \circ \eta)(\boldsymbol{x})$, hence isomorphic. □

Proof of Theorem 3. ($\Rightarrow$). We first introduce some notation. We fix a surjective environment $\eta : \mathcal{V} \to E_1$. Then, given an event $e \in E_1$, we write x_e to denote a fixed distinguished variable such that $\eta(x_e) = e$. Similarly, for a configuration $C = \{e_1, \dots, e_n\}$, we denote by $\boldsymbol{a}_C\boldsymbol{x}_C$ the corresponding tuple of labelled variables $(l(e_1)x_{e_1}, \dots, l(e_n)x_{e_n})$. Then, note that clearly η is $\boldsymbol{a}_C\boldsymbol{x}_C$-compatible.

Assuming $\mathcal{E}_1 \equiv_{\mathcal{L}_{wh}} \mathcal{E}_2$, we now show that the relation $R \subseteq Conf(\mathcal{E}_1) \times Conf(\mathcal{E}_2)$ defined by

$$\begin{aligned} R = \{(C_1, C_2) \mid\ & poset(C_1) \cong poset(C_2) \wedge \forall \varphi_{\boldsymbol{a}_{C_1}\boldsymbol{x}_{C_1}} \forall f : C_1 \to C_2 \text{ isomorphism.} \\ & \mathcal{E}_1, \varnothing \models_\eta \varphi_{\boldsymbol{a}_{C_1}\boldsymbol{x}_{C_1}} \text{ iff } \mathcal{E}_2, \varnothing \models_{f\circ\eta} \varphi_{\boldsymbol{a}_{C_1}\boldsymbol{x}_{C_1}}\} \end{aligned}$$

is a weak h-bisimulation between $\mathcal{E}_1$ and $\mathcal{E}_2$.

We proceed by contradiction. Assume that $(C_1, C_2) \in R$ and, without loss of generality, that $C_1 \xrightarrow{l(e)} C_1'$, but for all $e' \in E_2$ such that $l(e') = l(e)$ and $C_2 \xrightarrow{l(e')} C_2'$, we have $(C_1', C_2') \notin R$. This can happen either because $poset(C_1') \ncong poset(C_2')$ or because there exists a formula $\varphi_{\boldsymbol{a}_{C_1'}\boldsymbol{x}_{C_1'}}$ and an isomorphism $f : C_1' \to C_2'$ such that $\mathcal{E}_1, \varnothing \models_\eta \varphi_{\boldsymbol{a}_{C_1'}\boldsymbol{x}_{C_1'}}$ and $\mathcal{E}_2, \varnothing \not\models_{f\circ\eta} \varphi_{\boldsymbol{a}_{C_1'}\boldsymbol{x}_{C_1'}}$ (or vice versa, that is analogous, and so omitted).

Note that there must be at least one such transition $C_2 \xrightarrow{l(e')} C_2'$, labelled $l(e') = l(e)$, otherwise we would have $\mathcal{E}_1, \varnothing \models_\eta (l(e)\, x_e\rangle_{\boldsymbol{x}_{C_1}} \mathsf{T}$ and $\mathcal{E}_2, \varnothing \not\models_{f\circ\eta} (l(e)\, x_e\rangle_{\boldsymbol{x}_{C_1}} \mathsf{T}$ for any isomorphism $f : C_1 \to C_2$, contradicting the fact that $(C_1, C_2) \in R$.

Furthermore, since by hypothesis $\mathcal{E}_1$ and $\mathcal{E}_2$ are image-finite, there are finitely many transitions $C_2 \xrightarrow{l(e_i)} C_2^i$, for $i \in \{1, \dots, h\}$, complying with the previous conditions. Then, for each $i \in \{1, \dots, h\}$, by hypothesis we know that either $poset(C_1') \ncong poset(C_2^i)$ or there are a formula $\psi^i_{\boldsymbol{a}_{C_1'}\boldsymbol{x}_{C_1'}}$ and an isomorphism $f_i : C_1' \to C_2^i$ such that $\mathcal{E}_1, \varnothing \models_\eta \psi^i_{\boldsymbol{a}_{C_1'}\boldsymbol{x}_{C_1'}}$ and $\mathcal{E}_2, \varnothing \not\models_{f_i\circ\eta} \psi^i_{\boldsymbol{a}_{C_1'}\boldsymbol{x}_{C_1'}}$.

Split these configurations in two subsets, $A = \{C_2^i \,:\, poset(C_1') \ncong poset(C_2^i)\}$ and $B = \{C_2^i \,:\, poset(C_1') \cong poset(C_2^i)\}$. Then, for every $C_2^i \in A$, since $poset(C_1') \ncong poset(C_2^i)$ and η and $(f \circ \eta)[x_e \mapsto e^i]$ are $\boldsymbol{a}_{C_1'}\boldsymbol{x}_{C_1'}$-compatible, for any isomorphism $f : C_1 \to C_2$, by Lemma 6 we know that there must exist a formula $\theta^i_{\boldsymbol{a}_{C_1'}\boldsymbol{x}_{C_1'}}$ of the form $\bigvee_{\substack{\sigma : \boldsymbol{a}_{C_1'}\boldsymbol{x}_{C_1'} \to \boldsymbol{a}_{C_1'}\boldsymbol{x}_{C_1'} \\ \text{automorphism}}} \xi_{\boldsymbol{x}_{C_1'}}\sigma$ such that $\mathcal{E}_1, \varnothing \models_\eta \theta^i_{\boldsymbol{a}_{C_1'}\boldsymbol{x}_{C_1'}}$ and $\mathcal{E}_2, \varnothing \not\models_{(f\circ\eta)[x_e\mapsto e^i]} \theta^i_{\boldsymbol{a}_{C_1'}\boldsymbol{x}_{C_1'}}$. Now consider the formula

$$\varphi_{\boldsymbol{a}_{C_1'}\boldsymbol{x}_{C_1'}} = (l(e)\, x_e\rangle_{\boldsymbol{x}_{C_1}} \left(\bigwedge_{i\,:\,C_2^i \in A} \theta^i_{\boldsymbol{a}_{C_1'}\boldsymbol{x}_{C_1'}} \;\wedge\; \bigwedge_{i\,:\,C_2^i \in B} \psi^i_{\boldsymbol{a}_{C_1'}\boldsymbol{x}_{C_1'}} \right)$$

By hypothesis it is easy to see that $\mathcal{E}_1, \varnothing \models_\eta \varphi_{\boldsymbol{a}_{C_1'}\boldsymbol{x}_{C_1'}}$. However, for any isomorphism $f : C_1 \to C_2$

- for every i s.t. $C_2^i \in A$, we know that $\mathcal{E}_2, \varnothing \not\models_{(f \circ \eta)[x_e \mapsto e^i]} \theta^i_{\boldsymbol{a}_{C_1'}\boldsymbol{x}_{C_1'}}$; and
- for every i s.t. $C_2^i \in B$, we know that $\mathcal{E}_2, \varnothing \not\models_{f_i \circ \eta} \psi^i_{\boldsymbol{a}_{C_1'}\boldsymbol{x}_{C_1'}}$, and so, taking $\sigma = (f \cup \{(e, e_i)\}) \circ f_i^{-1}$, hence $\sigma \circ f_i \circ \eta = (f \circ \eta)[x_e \mapsto e_i]$, since $f_i \circ \eta$ and $(f \circ \eta)[x_e \mapsto e_i]$ are both $\boldsymbol{a}_{C_1'}\boldsymbol{x}_{C_1'}$-compatible, by Lemma 7 we have that $\mathcal{E}_2, \varnothing \not\models_{(f \circ \eta)[x_e \mapsto e_i]} \psi^i_{\boldsymbol{a}_{C_1'}\boldsymbol{x}_{C_1'}}$.

Therefore, we would have that $\mathcal{E}_2, \varnothing \not\models_{f \circ \eta} \varphi_{\boldsymbol{a}_{C_1}\boldsymbol{x}_{C_1}}$ contradicting the fact that $(C_1, C_2) \in R$. Thus, we can conclude that R is a weak h-bisimulation by observing that $(\varnothing, \varnothing) \in R$, since clearly $\mathit{poset}(\varnothing) \cong \mathit{poset}(\varnothing)$ and by hypothesis $\mathcal{E}_1 \equiv_{\mathcal{L}_{wh}} \mathcal{E}_2$.

($\Leftarrow$). Assume we have a weak h-bisimulation R between $\mathcal{E}_1$ and $\mathcal{E}_2$. We prove that $\mathcal{E}_1 \equiv_{\mathcal{L}_{wh}} \mathcal{E}_2$, i.e., they satisfy the same ϵ-parameterised formulae. Actually, we show that, for every pair of configurations $(C_1, C_2) \in R$ (hence isomorphic), for every formula $\varphi_{\boldsymbol{ax}}$ and for every pair of $\boldsymbol{ax}$-compatible environments $\eta_1 \in \mathit{Env}_{\mathcal{E}_1}$ and $\eta_2 \in \mathit{Env}_{\mathcal{E}_2}$ such that $\eta_1(\boldsymbol{x}) = C_1$ and $\eta_2(\boldsymbol{x}) = C_2$, it holds $\mathcal{E}_1, \varnothing \models_{\eta_1} \varphi_{\boldsymbol{ax}}$ if and only if $\mathcal{E}_2, \varnothing \models_{\eta_2} \varphi_{\boldsymbol{ax}}$. Observing that environments are irrelevant when $\boldsymbol{x} = \epsilon$, this is enough since by hypothesis $(\varnothing, \varnothing) \in R$, implying that the two PESs would satisfy the same ϵ-parameterised formulae, i.e., the same (closed) formulae of $\mathcal{L}_{wh}$.

We proceed by induction on the shape of the formula $\varphi_{\boldsymbol{ax}}$. We discuss only some cases and a single direction, the other being symmetric.

$\varphi_{\boldsymbol{ax}} = \neg\,\psi_{\boldsymbol{ax}}$**:** Assume that $\mathcal{E}_1, \varnothing \models_{\eta_1} \neg\,\psi_{\boldsymbol{ax}}$. By definition of the semantics, we know that $\mathcal{E}_1, \varnothing \not\models_{\eta_1} \psi_{\boldsymbol{ax}}$. Then, since $\eta_1(\boldsymbol{x}) = C_1$ and $\eta_2(\boldsymbol{x}) = C_2$, by inductive hypothesis we have that $\mathcal{E}_2, \varnothing \not\models_{\eta_2} \psi_{\boldsymbol{ax}}$. Again by definition of the semantics, we conclude that $\mathcal{E}_2, \varnothing \models_{\eta_2} \neg\,\psi_{\boldsymbol{ax}}$.

$\varphi_{\boldsymbol{ax}} = (b\,z\rangle_{\boldsymbol{x}}\,\psi_{\boldsymbol{a}'\boldsymbol{x}'}$**, where** $\boldsymbol{a}'\boldsymbol{x}' = \boldsymbol{ax}, bz$**:** Assume that $\mathcal{E}_1, \varnothing \models_{\eta_1} (b\,z\rangle_{\boldsymbol{x}}\,\psi_{\boldsymbol{a}'\boldsymbol{x}'}$. By definition of the semantics we know that there exists an event $e \in E$ such that $l(e) = b$, $\eta_1(\boldsymbol{x}) = C_1 \xrightarrow{b} C_1' = C_1 \cup \{e\} = \eta_1[z \mapsto e](\boldsymbol{x}')$ and $\mathcal{E}_1, \varnothing \models_{\eta_1[z \mapsto e]} \psi_{\boldsymbol{a}'\boldsymbol{x}'}$.

Since $(C_1, C_2) \in R$, there must also exist a transition $\eta_2(\boldsymbol{x}) = C_2 \xrightarrow{b} C_2' = C_2 \cup \{e'\}$, for some event e', such that $l(e') = l(e) = b$ and $\mathit{poset}(C_1') \cong \mathit{poset}(C_2')$. Then, since $\eta_1[z \mapsto e]$ and $\eta_2[z \mapsto e']$ are $\boldsymbol{a}'\boldsymbol{x}'$-compatible, by inductive hypothesis we have that $\mathcal{E}_2, \varnothing \models_{\eta_2[z \mapsto e']} \psi_{\boldsymbol{a}'\boldsymbol{x}'}$. Again by definition of the semantics, we can conclude that $\mathcal{E}_2, \varnothing \models_{\eta_2} (b\,z\rangle_{\boldsymbol{x}}\,\psi_{\boldsymbol{a}'\boldsymbol{x}'}$.

$\varphi_{\boldsymbol{ax}} = \bigvee_{\substack{\sigma : \boldsymbol{ax} \to \boldsymbol{ax} \\ \textbf{automorphism}}} \xi_{\boldsymbol{x}}\sigma$**:** Immediate by Lemma 6, since by hypothesis we know that η_1 and η_2 are $\boldsymbol{ax}$-compatible, $\mathit{poset}(C_1) \cong \mathit{poset}(C_2)$, $\eta_1(\boldsymbol{x}) = C_1$ and $\eta_2(\boldsymbol{x}) = C_2$. □

References

[1] P. Baldan, A. Carraro, A causal view on non-interference, Fundam. Inform. 140 (1) (2015) 1–38.

[2] P. Baldan, S. Crafa, Hereditary history-preserving bisimilarity: logics and automata, in: J. Garrigue (Ed.), Proceedings of APLAS'14, in: LNCS, vol. 8858, Springer, 2014, pp. 469–488.

[3] P. Baldan, S. Crafa, A logic for true concurrency, J. ACM 61 (4) (2014) 24:1–24:36.

[4] P. Baldan, D. Gorla, T. Padoan, I. Salvo, Behavioural logics for configuration structures, Theor. Comput. Sci. (2021), submitted for publication.

[5] M.A. Bednarczyk, Hereditary history preserving bisimulations or what is the power of the future perfect in program logics, Technical report, Polish Academy of Sciences, 1991.

[6] G. Boudol, I. Castellani, On the semantics of concurrency: partial orders and transition systems, in: Proc. of TAPSOFT'87, 1987, pp. 123–137.

[7] G. Boudol, I. Castellani, Concurrency and atomicity, Theor. Comput. Sci. 59 (1988) 25–84.

[8] J. Bradfield, S. Fröschle, Independence-friendly modal logic and true concurrency, Nord. J. Comput. 9 (1) (2002) 102–117.

[9] A. Bucciarelli, T. Ehrhard, Extensional embedding of a strongly stable model of PCF, in: Proc. of ICALP, in: LNCS, vol. 510, Springer, 1991, pp. 35–46.

[10] A. Bucciarelli, T. Ehrhard, Sequentiality and strong stability, in: Proc. of LICS, IEEE Computer Society, 1991, pp. 138–145.

[11] I. Castellani, Process algebras with localities, in: J. Bergstra, A. Ponse, S. Smolka (Eds.), Handbook of Process Algebra, Elsevier, 2001, pp. 945–1045, Chapter 15.

[12] L. Castellano, G.D. Michelis, L. Pomello, Concurrency versus interleaving: an instructive example, Bull. Eur. Assoc. Theor. Comput. Sci. 31 (1987) 12–14.

[13] S. Chakraborty, V. Vafeiadis, Grounding thin-air reads with event structures, in: POPL 2019, in: PACMPL, vol. 3, ACM, 2019, pp. 70:1–70:28.

[14] S. Christensen, Decidability and decomposition in process algebra, PhD thesis, University of Edinburgh, 1993.

[15] R. De Nicola, G. Ferrari, Observational logics and concurrency models, in: K.V. Nori, C.E.V. Madhavan (Eds.), Proc. of FST-TCS'90, in: LNCS, vol. 472, Springer, 1990, pp. 301–315.

[16] R. De Nicola, U. Montanari, F.W. Vaandrager, Back and forth bisimulations, in: Proc. of CONCUR, in: LNCS, vol. 458, Springer, 1990, pp. 152–165.

[17] P. Degano, R. De Nicola, U. Montanari, Observational equivalences for concurrency models, in: Formal Description of Programming Concepts - III, North-Holland, 1987, pp. 105–134.

[18] P. Degano, R. De Nicola, U. Montanari, Partial orderings descriptions and observations of nondeterministic concurrent processes, in: Linear Time, Branching Time and Partial Order in Logics and Models for Concurrency, in: LNCS, vol. 354, Springer, 1988, pp. 438–466.

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

[20] A. Farzan, P. Madhusudan, Causal atomicity, in: T. Ball, R.B. Jones (Eds.), Proceedings of CAV'06, in: LNCS, vol. 4144, 2006, pp. 315–328.

[21] H. Fecher, A completed hierarchy of true concurrent equivalences, Inf. Process. Lett. 89 (5) (2004) 261–265.

[22] S. Fröschle, Decidability and Coincidence of Equivalences for Concurrency, PhD thesis, Univeristy of Edinburgh, 2004.

[23] S. Fröschle, S. Lasota, Decomposition and complexity of hereditary history preserving bisimulation on BPP, in: CONCUR'05, in: LNCS, vol. 3653, Springer, 2005, pp. 263–277.

[24] S. Fröschle, P. Jančar, S. Lasota, Z. Sawa, Non-interleaving bisimulation equivalences on basic parallel processes, Inf. Comput. 208 (1) (2010) 42–62.

[25] J. Girard, Linear logic, Theor. Comput. Sci. 50 (1987) 1–102.

[26] D. Gorla, I. Salvo, Conflict vs causality in event structures, J. Log. Algebraic Methods Program. 119 (2021). Full and revised version of [27].

[27] D. Gorla, I. Salvo, A. Piperno, Conflict vs causality in event structures, in: J.A. Pérez, J. Rot (Eds.), Proceedings of EXPRESS/SOS 2019, Amsterdam, 2019, in: EPTCS, vol. 300, 2019, pp. 86–101.

[28] R. Gorrieri, Process Algebras for Petri Nets: The Alphabetization of Distributed Systems, EATCS Monographs in Theoretical Computer Science, Springer, 2017.

[29] R. Gorrieri, A study on team bisimulations for BPP nets, in: R. Janicki, N. Sidorova, T. Chatain (Eds.), Application and Theory of Petri Nets and Concurrency, Springer, 2020, pp. 153–175.

[30] R. Gorrieri, Team bisimilarity, and its associated modal logic, for BPP nets, Acta Inform. 58 (5) (2021) 529–569.

[31] J. Gutierrez, Logics and bisimulation games for concurrency, causality and conflict, in: L. de Alfaro (Ed.), Proc. of FoSSaCS'09, in: LNCS, vol. 5504, Springer, 2009, pp. 48–62.

[32] J. Gutierrez, On bisimulation and model-checking for concurrent systems with partial order semantics, PhD thesis, University of Edinburgh, 2011.

[33] J. Gutierrez, J.C. Bradfield, Model-checking games for fixpoint logics with partial order models, in: M. Bravetti, G. Zavattaro (Eds.), Proc. of CONCUR'09, in: LNCS, vol. 5710, Springer, 2009, pp. 354–368.

[34] M. Hennessy, R. Milner, Algebraic laws for nondeterminism and concurrency, J. ACM 32 (1) (1985) 137–161.

[35] C.A.R. Hoare, Communicating sequential processes, Commun. ACM 21 (8) (1978) 666–677.

[36] 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.

[37] K. Krukow, M. Nielsen, Trust structures, Int. J. Inf. Secur. 6 (2–3) (2007) 153–181.

[38] K. Krukow, M. Nielsen, V. Sassone, A framework for concrete reputation-systems with applications to history-based access control, in: Proc. of CCS, ACM, 2005, pp. 260–269.

[39] R. Milner, Communication and Concurrency, Prentice Hall, 1989.

[40] M. Nielsen, C. Clausen, Games and logics for a noninterleaving bisimulation, Nord. J. Comput. 2 (2) (1995) 221–249.

[41] M. Nielsen, K. Krukow, Transfer of trust in event-based reputation systems, Theor. Comput. Sci. 429 (2012) 236–246.

[42] M. Nielsen, G.D. Plotkin, G. Winskel, Petri nets, event structures and domains, part I, Theor. Comput. Sci. 13 (1981) 85–108.

[43] M. Nielsen, P.S. Thiagarajan, Regular event structures and finite petri nets: the conflict-free case, in: Applications and Theory of Petri Nets, in: LNCS, vol. 2360, Springer, 2002, pp. 335–351.

[44] W. Penczek, Branching time and partial order in temporal logics, in: Time and Logic: A Computational Approach, UCL Press, 1995, pp. 179–228.

[45] I. Phillips, I. Ulidowski, A logic with reverse modalities for history-preserving bisimulations, Electron. Proc. Theor. Comput. Sci. 64 (2011) 104–118.

[46] I. Phillips, I. Ulidowski, Event identifier logic, Math. Struct. Comput. Sci. 24 (2) (2014) 1–51.

[47] 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.

[48] S. Pinchinat, F. Laroussinie, P. Schnoebelen, Logical characterization of truly concurrent bisimulation, Technical Report 114, LIFIA-IMAG, Grenoble, 1994.

[49] L. Pomello, Some equivalence notions for concurrent systems. An overview, in: Advances in Petri Nets, in: LNCS, vol. 222, Springer, 1985, pp. 381–400.

[50] A. Rabinovich, B.A. Trakhtenbrot, Behaviour structures and nets, Fundam. Inform. 11 (4) (1988) 357–404.

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

[52] R.J. van Glabbeek, U. Goltz, J. Schicke-Uffmann, On characterising distributability, Log. Methods Comput. Sci. 9 (3) (2013).

[53] R.J. van Glabbeek, G.D. Plotkin, Configuration structures, in: Proc. of LICS, IEEE Computer Society, 1995, pp. 199–209.

[54] R.J. van Glabbeek, F.W. Vaandrager, Petri net models for algebraic theories of concurrency, in: Proceedings of PARLE, Parallel Architectures and Languages Europe, Volume II, in: Lecture Notes in Computer Science, vol. 259, Springer, 1987, pp. 224–242.

[55] G. Winskel, Event structures, in: W. Brauer, W. Reisig, G. Rozenberg (Eds.), Petri Nets: Applications and Relationships to Other Models of Concurrency, in: LNCS, vol. 255, Springer, 1987, pp. 325–392.

[56] G. Winskel, Events, causality and symmetry, Comput. J. 54 (1) (2011) 42–57.

[57] G. Winskel, M. Nielsen, Models for concurrency, in: S. Abramsky, D.M. Gabbay, T.S.E. Maibaum (Eds.), Handbook of Logic in Computer Science, Vol. 4, Oxford University Press, Inc., New York, NY, USA, 1995, pp. 1–148.