Math. Struct. in Comp. Science (2014), vol. 24, iss. 2, e240204, 51 pages.
doi:10.1017/S0960129513000510

Event Identifier Logic[†]

IAIN PHILLIPS[‡] and IREK ULIDOWSKI[§¶]

[‡]*Department of Computing, Imperial College London,*
180 Queen's Gate, London SW7 2AZ, United Kingdom
Email: iccp@doc.ic.ac.uk
[§]*Department of Computer Science, University of Leicester,*
University Road, Leicester LE1 7RH, United Kingdom
Email: iu3@mcs.le.ac.uk

Received 20 January 2012; revised 5 November 2012

In this paper we introduce Event Identifier Logic (EIL), which extends Hennessy–Milner logic by the addition of:

(1) reverse as well as forward modalities; and
(2) identifiers to keep track of events.

We show that this logic corresponds to hereditary history-preserving (HH) bisimulation equivalence within a particular true-concurrency model, namely, stable configuration structures. We also show how natural sublogics of EIL correspond to coarser equivalences. In particular, we provide logical characterisations of weak-history- preserving (WH) and history-preserving (H) bisimulation. Logics corresponding to HH and H bisimulation have been given previously, but none, as far as we are aware, corresponding to WH bisimulation (when autoconcurrency is allowed). We also present characteristic formulas that characterise individual structures with respect to history-preserving equivalences.

1. Introduction

In this paper we present a modal logic that can express simple properties of computation in the true concurrency setting of stable configuration structures. We aim, like Hennessy–Milner logic (HML) (Hennessy and Milner 1985) in the interleaving setting, to characterise the main true concurrency equivalences and to develop characteristic formulas for them.

HML has a 'diamond' modality $\langle a \rangle \phi$, which says that an event labelled a can be performed, taking us to a new state that satisfies ϕ. The logic also contains negation ($\neg$), conjunction ($\wedge$) and a base formula that always holds ($\mathtt{tt}$). HML is strong enough to distinguish any two processes that are not bisimilar.

We are interested in making true concurrency distinctions between processes. These processes will be *event structures*, where the current state is represented by the set of

[†] This paper was originally submitted for inclusion in the forthcoming EXPRESS 2011 Special Issue of *Mathematical Structures in Computer Science*. It was accepted, and the revised (final) version was sent in November 2012. Subsequently, in April 2013, in view of the extra time required to handle other articles in the Special Issue, the Guest Editors very kindly agreed to the article being published separately in the Journal to avoid further delay in publication.

[¶] Irek Ulidowski is grateful for partial support provided by JSPS grants S-09053 and FU-019, and by Nagoya University and the Research Institute for Mathematical Sciences, Kyoto University during a research visit.

events that have already occurred. Such sets are called *configurations*. Events have labels (ranged over by $a, b, \ldots$), and different events may have the same label. We shall refer to example event structures using a CCS-like notation, with:

— $a \mid b$ denoting an event labelled with a in parallel with another event labelled with b;
— $a.b$ denoting two events labelled a and b where the first causes the second; and
— $a + b$ denoting two events labelled a and b that conflict.

In the true concurrency setting, bisimulation is referred to as *interleaving bisimulation*, or IB for short. The processes $a \mid b$ and $a.b + b.a$ are interleaving bisimilar, but from the point of view of true concurrency they should be distinguished, and HML is not powerful enough to do this.

We therefore look for a more powerful logic, and we base this logic on the addition of reverse moves. Instead of the single modality $\langle a \rangle \phi$, we shall now have two:

— *forward diamond* $\langle a \rangle\rangle \phi$, which is just a new notation for the $\langle a \rangle \phi$ of HML; and
— *reverse diamond* $\langle\langle a \rangle \phi$.

The latter is satisfied if we can reverse some event labelled with a and get to a configuration where ϕ holds. Such an event would have to be *maximal* to enable us to reverse it, that is, it could not be the cause of some other event that has already occurred.

With this new reverse modality, we can now distinguish $a \mid b$ and $a.b + b.a$ since $a \mid b$ satisfies $\langle a \rangle\rangle \langle b \rangle\rangle \langle\langle a \rangle \mathrm{tt}$, while $a.b + b.a$ does not. The formula expresses the idea that a and b are *concurrent*. Alternatively, we can see that $a.b + b.a$ satisfies $\langle a \rangle\rangle \langle b \rangle\rangle \neg \langle\langle a \rangle \mathrm{tt}$, while $a \mid b$ does not. This latter formula expresses the idea that *a causes b*.

The new logic corresponds to *reverse interleaving bisimulation* (Phillips and Ulidowski 2012), or RI-IB for short. In the absence of autoconcurrency, Bednarczyk showed that this is as strong as *hereditary history-preserving bisimulation* (Bednarczyk 1991), or HH for short, which is usually regarded as the strongest desirable true concurrency equivalence. HH was independently proposed in Joyal *et al.* (1996) under the name of strong history-preserving bisimulation.

Autoconcurrency is where events can occur concurrently and have the same label. To allow for this, we need to strengthen the logic. For instance, we want to distinguish $a \mid a$ from $a.a$, which is not possible with the logic as it stands since $\langle a \rangle\rangle \langle a \rangle\rangle \langle\langle a \rangle \mathrm{tt}$ is satisfied by both processes. We need some way of distinguishing the two events labelled with a. To achieve this, we change our modalities so that when we make a forward move we *declare* an *identifier* (ranged over by $x, y, \ldots$) that stands for that event, which allows us to refer to it again when reversing it. Now we can write $\langle x : a \rangle\rangle \langle y : a \rangle\rangle \langle\langle x \rangle \mathrm{tt}$, and this is satisfied by $a \mid a$ but not by $a.a$. Declaration is an identifier-binding operation, so x and y are both bound in the formula. Baldan and Crafa (2010) also used such declarations in their forward-only logic.

With this simple change, we now have a logic that is as strong as HH, even with autoconcurrency.

However, we have to be careful that our logic does not become too strong. For instance, we want to ensure that processes a and $a + a$ are indistinguishable. One might think that $a + a$ satisfies $\langle x : a \rangle\rangle \langle\langle x \rangle \langle y : a \rangle\rangle \neg \langle\langle x \rangle \mathrm{tt}$, while a does not. To avoid this, we need to ensure that x is forgotten once it is reversed so that it cannot be used again. One could make a syntactic restriction saying that in a formula $\langle\langle x\rangle\phi$, the identifier x is not allowed to occur (free) in ϕ. However, this is not actually necessary since our semantics will ensure that all identifiers must be assigned to events in the current configuration. So, in fact,

$$\langle x : a\rangle\rangle\langle\langle x\rangle\langle y : a\rangle\rangle\neg\langle\langle x\rangle\mathrm{tt}$$

is not satisfied by $a + a$, since we are not allowed to reverse x as it would take us to a configuration where x is mentioned in $\langle y : a\rangle\rangle\neg\langle\langle x\rangle\mathrm{tt}$ but x is assigned to an event outside the current configuration. Baldan and Crafa also had to deal with this issue.

Our logic is not quite complete, since we wish to express certain further properties. For instance, we would like to express a reverse move labelled with a, that is, $\langle\langle a\rangle\phi$. Instead of adding this directly, we add *declarations* $(x : a)\phi$. We can now express $\langle\langle a\rangle\phi$ using the formula $(x : a)\langle\langle x\rangle\phi$ (where x does not occur (free) in ϕ).

We also wish to express so-called *step transitions*, which are transitions consisting of multiple events occurring concurrently. For instance, a forward step $\langle a, a\rangle\rangle\phi$ of two events labelled with a can be achieved by

$$\langle x : a\rangle\rangle\langle y : a\rangle\rangle(\phi \wedge \langle\langle x\rangle\mathrm{tt})$$

and a reverse step $\langle\langle a, a\rangle\phi$ can be achieved by

$$(x : a)(y : a)(\langle\langle x\rangle\langle\langle y\rangle\phi \wedge \langle\langle y\rangle\mathrm{tt})$$

(both formulas with x and y not free in ϕ). Thus the reverse steps employ declarations. As well as expressing reverse steps, declarations allow us to obtain a sublogic corresponding to *weak history-preserving bisimulation* (WH).

This completes a brief introduction of our logic, which we call *Event Identifier Logic*, or EIL for short. Apart from corresponding to HH, EIL has natural sublogics for several other true concurrency equivalences. Figure 1 shows a hierarchy of equivalences that we are able to characterise, where arrows denote proper set inclusion. Apart from the HH and WH already mentioned, *history-preserving bisimulation* (H) is a widely studied equivalence that employs history isomorphism. *Hereditary weak-history preserving bisimulation* (HWH) is WH with the hereditary property (Bednarczyk 1991) that deals with the reversing of events. We also consider *pomset bisimulation* PB (where transitions are pomsets), *step bisimulation* SB (where transitions are 'steps', that is, sets of concurrent events), and the combination of WH and PB, namely WHPB. The definitions of these equivalences can be found in van Glabbeek and Goltz (2001) and Phillips and Ulidowski (2012), and are outlined in Section 3.2 of the current paper.

It is natural to ask whether, at least for a finite structure, there is a single logical formula that captures all of its behaviour, up to a certain equivalence. Such formulas are called *characteristic* formulas. They have been investigated previously for HML and other logics (Graf and Sifakis 1986; Steffen and Ingólfsdóttir 1994; Aceto *et al.* 2009). We shall look at characteristic formulas with respect to three of the equivalences we consider, namely, HH, H and WH. As far as we are aware, this is the first time that characteristic formulas have been investigated in the true concurrency setting.

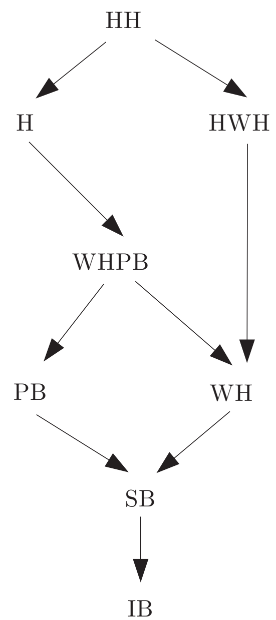

Fig. 1. The hierarchy of equivalences considered in this paper.

The main contribution of the paper is a logic EIL. It could be argued that EIL is a natural and canonical logic for the true concurrency equivalences considered here in the following sense:

(1) The forward and reverse modalities faithfully capture the information of the forward and reverse transitions in the definitions of the equivalences, particularly in the case of the history-preserving equivalences.
(2) Event identifier environments and event declarations give rise naturally to order isomorphisms for HH, H, HWH and WH.
(3) EIL extends HML and keeps with its spirit of having simple modalities defined seamlessly over a general computation model.

Other contributions of the paper include what we believe to be the first logics for WH and HWH (and also WHPB). We also give a full proof of EIL's characterisation of HH in the presence of autoconcurrency. Finally, we present what we believe to be the first characteristic formulas for HH, H and WH.

1.1. *Organisation of the paper*

We look at related work in Section 2. Then, in Section 3, we recall the definitions of configuration structures and the bisimulation-based equivalences that we shall need. We then introduce EIL in Section 4, giving examples of its usage. In Section 5, we look at how to characterise various equivalences using EIL and its sublogics, and then, in Section 6, we investigate characteristic formulas. Finally, we present our conclusions and some suggestions for future work in Section 7.

Remark 1.1. The current paper extends the preliminary version Phillips and Ulidowski (2011) through the inclusion of full proofs of all results, the addition of sublogics for further equivalences, including pomset bisimulation and step bisimulation (Section 5.3) and the inclusion of more examples.

2. Related work

Previous work on logics for true concurrency can be categorised loosely according to the type of semantic structure (model) that the satisfaction relation of the logic is defined for. There are logics over configurations (sets of consistent events) (Goltz *et al.* 1992; Baldan and Crafa 2010) and logics over paths (or computations) (Cherief 1992; Nielsen and Clausen 1994a; Nielsen and Clausen 1994b; Nielsen and Clausen 1995; Pinchinat *et al.* 1994), although the logics in Nielsen and Clausen (1994a), Nielsen and Clausen (1994b) and Nielsen and Clausen (1995) can also be viewed as logics over configurations. Other structures such as trees, graphs and Kripke frames are used as models in, for example, De Nicola and Ferrari (1990), Mukund and Thiagarajan (1992), Gutierrez (2009) and Gutierrez and Bradfield (2009).

The logic in the current paper uses simple forward and reverse event identifier modalities that are sufficient to characterise HH. In contrast, Baldan and Crafa (2010; 2011) achieved an alternative characterisation of HH with a different modal logic that only uses forward-only event identifier modalities $\langle x \rangle$ and $(\boldsymbol{x}, \bar{\boldsymbol{y}} < \mathsf{a}\, z)$. The formula $(\boldsymbol{x}, \bar{\boldsymbol{y}} < \mathsf{a}\, z)\phi$ holds in a configuration if in its future there is an a-labelled event e that can be bound to z, and ϕ holds. Additionally, e must be:

(1) caused at least by the events already bound to the events in $\boldsymbol{x}$; and
(2) concurrent with at least the events already bound to the events in $\boldsymbol{y}$.

Baldan and Crafa (2010) also identified several interesting sublogics characterising H, pomset bisimulation (Boudol and Castellani 1987; van Glabbeek and Goltz 2001) and step bisimulation (Pomello 1986; van Glabbeek and Goltz 2001). Baldan and Crafa also proposed an extension of the logic with recursion in order to be able to describe certain properties of infinite computations (Baldan and Crafa 2011).

Goltz, Kuiper and Penczek (Goltz *et al.* 1992) researched configurations of prime event structures *without autoconcurrency*. In such a setting, HH coincides with reverse interleaving bisimulation RI-IB (Phillips and Ulidowski 2006; Phillips and Ulidowski 2007; Phillips and Ulidowski 2012) – this was shown in Bednarczyk (1991). Moreover, H coincides with WH. *Partial Order Logic* (POL), which was proposed in Goltz *et al.* (1992), contains past modalities, and the authors stated that it characterises RI-IB (and thus HH). It is also conjectured that if POL is restricted in such a way that no forward modalities can be nested in a past modality, then such a logic characterises H (and thus WH).

Cherief (1992) defined a pomset bisimulation relation over paths and showed that it coincides with H (defined over configurations). The author then predicted that an extension of HML with forward and reverse pomset modalities characterises H. This idea was then developed further by Pinchinat, Laroussinie and Schnoebelen in Pinchinat *et al.* (1994).

Nielsen and Clausen defined a δ-bisimulation relation (δb) over paths (Nielsen and Clausen 1994a; Nielsen and Clausen 1995). However, unlike the case for the relation in Cherief (1992) and Pinchinat *et al.* (1994), independent maximal events can be reversed in any order. This seemingly small change has a profound effect on the strength of the equivalence since δb coincides with HH. It has been shown that an extension of HML with a reverse modality characterises HH when there is no autoconcurrency (Nielsen and Clausen 1994a; Nielsen and Clausen 1995). Additionally, Nielsen and Clausen (1994b) stated (without a proof) that an extension of HML with a reverse *event index* modality characterises HH even in the presence of autoconcurrency. The notion of paths used in Nielsen and Clausen (1994a), Nielsen and Clausen (1994b) and Nielsen and Clausen (1995) induces a notion of configuration, so their logics could be understood as logics over configurations, and their reverse index modality could be seen as a form of our reverse event identifier modality. We would argue, however, that many properties of configurations related to causality and concurrency between events are expressed more naturally using reverse identifier modalities.

Past or reverse modalities, which are central to our logic, have already been used in a number of modal logics and temporal logics (Hennessy and Stirling 1985; De Nicola and Vaandrager 1990; De Nicola *et al.* 1990; De Nicola and Ferrari 1990; Goltz *et al.* 1992; Laroussinie *et al.* 1995; Laroussinie and Schnoebelen 1995; Penczek 1995), but only De Nicola and Ferrari (1990) and Goltz *et al.* (1992) proposed logical characterisations of true concurrency equivalences. By contrast, the HML with backward modalities in De Nicola and Vaandrager (1990) and De Nicola *et al.* (1990) defined over paths is shown to characterise branching bisimulation. Finally, Gutierrez introduced a modal logic for transition systems with independence (Gutierrez 2009; Gutierrez and Bradfield 2009) that has two diamond modalities: one for causally dependent transitions and the other for concurrent transitions with respect to a given transition.

3. Configuration structures and equivalences

In this section we shall define our computational model (stable configuration structures) and the various bisimulation equivalences for which we shall present logical characterisations.

3.1. *Configuration structures*

We shall work with stable configuration structures (van Glabbeek and Plotkin 1995; van Glabbeek and Plotkin 2009; van Glabbeek and Goltz 2001), which are equivalent to stable event structures (Winskel 1987).

However, we shall first recall the definition of prime event structures, which are better known than configuration structures, and which will be useful, along with CCS expressions, for expressing many of our examples. A prime event structure is a set of events with a labelling function, together with a causality relation and a conflict relation (between events that cannot be members of the same configuration). We assume a set of action labels Act, ranged over by $a, b, \ldots$.

Definition 3.1 (Nielsen *et al.* 1981). A (labelled) prime event structure is a 4-tuple $\mathcal{E} = (E, <, \sharp, \ell)$ where:

— E is a set of *events*;
— $< \subseteq E \times E$ is an irreflexive partial order (the *causality relation*) such that for any $e \in E$, the set $\{e' \in E : e' < e\}$ is finite;
— $\sharp \subseteq E \times E$ is an irreflexive, symmetric relation (the *conflict relation*) such that if $e_1 < e_2$ and $e_1 \sharp e$, then $e_2 \sharp e$;
— $\ell : E \to \mathsf{Act}$ is the *labelling function.*

Prime event structures as we have defined them have *binary* conflict, though this can also be generalised to non-binary conflict (van Glabbeek and Vaandrager 1997).

A *configuration* of a prime event structure is a finite set of events that is downwards-closed (left-closed) under the causal ordering $<$ and is conflict-free, that is, no two events can be related by $\sharp$. Thus, a configuration represents a possible state of a computation, being the set of all events that have happened so far.

Example 3.2. Consider a prime event structure with events e_1, e_2, e_3 all labelled with a, where e_1 causes e_2 and e_1, e_2 are concurrent with e_3. The corresponding CCS expression is $(a.a) \,|\, a$. The set of configurations consists of $\emptyset$, $\{e_1\}, \{e_3\}, \{e_1, e_2\}, \{e_1, e_3\}$ and $\{e_1, e_2, e_3\}$.

When drawing diagrams of prime event structures, we shall, as usual, depict the causal relation with arrows and the conflict relation with dotted lines. We shall also suppress the actual events and write their labels instead. Thus, if we have two events e_1 and e_2, both labelled with a, in diagrams we shall denote them both as a, or sometimes as a_1 and a_2, respectively, when we wish to distinguish between them. This is justified because all the forms of equivalence we shall discuss depend on the labels of the events rather than the events themselves.

We arrive at configuration structures by treating the configurations of an event structure as a first-class notion, rather than obtaining them from the causal and conflict relations on events.

Definition 3.3. A *configuration structure* (over an alphabet of labels Act) is a pair $\mathcal{C} = (C, \ell)$, where C is a family of finite sets (configurations) and $\ell : \bigcup_{X \in C} X \to \mathsf{Act}$ is a labelling function.

We use $C_{\mathcal{C}}, \ell_{\mathcal{C}}$ to refer to the two components of a configuration structure $\mathcal{C}$ and write $E_{\mathcal{C}} = \bigcup_{X \in C} X$ to denote the *events* of $\mathcal{C}$. We let $e, \ldots$ range over events, $E, F, \ldots$ range over sets of events and $X, Y, \ldots$ range over configurations. We let $a, b, c, \ldots$ range over labels in Act.

Definition 3.4 (van Glabbeek and Goltz 2001). A configuration structure $\mathcal{C} = (C, \ell)$ is *stable* if it is:

— rooted, that is, $\emptyset \in C$;
— connected, that is, $\emptyset \neq X \in C$ implies $\exists e \in X : X \setminus \{e\} \in C$;
— closed under bounded unions, that is, if $X, Y, Z \in C$, then $X \cup Y \subseteq Z$ implies $X \cup Y \in C$;

https://doi.org/10.1017/S0960129513000510 Published online by Cambridge University Press

— closed under bounded intersections, that is, if $X, Y, Z \in C$, then $X \cup Y \subseteq Z$ implies $X \cap Y \in C$.

The set of configurations of a prime event structure $\mathcal{E}$ forms a stable configuration structure. To see this, we can check the four conditions of Definition 3.4. It is clear that the empty set is always a configuration of $\mathcal{E}$. For connectedness, if X is a non-empty configuration of $\mathcal{E}$ and e is any maximal event in X, then $X \setminus \{e\}$ is also a configuration. If X and Y are configurations, then $X \cup Y$ is not necessarily a configuration. It is left-closed, but it may contain conflict between events of X and those of Y, so X and Y represent alternative and incompatible possible states of a computation. However, if $X \cup Y \subseteq Z$ for some configuration Z, then $X \cup Y$ is clearly conflict free, and thus a configuration of $\mathcal{E}$.

The most interesting condition is the last one. If X and Y are configurations, then $X \cap Y$ is a configuration of $\mathcal{E}$ since it is left-closed and conflict-free. Thus, for prime event structures, configurations are closed under arbitrary intersections and not just bounded intersections. This shows that stable configuration structures are more general than prime event structures. We require closure under intersections, but only between compatible configurations (the boundedness condition).

Example 3.5. Let a configuration structure $\mathcal{C}$ have the following configurations:

$$\varnothing \quad \{e_1\} \quad \{e_2\} \quad \{e_1, e_3\} \quad \{e_2, e_3\}$$

(we omit the labelling since it is not relevant here). It is easy to check that C satisfies the four conditions of Definition 3.4 and hence is stable. However, C is not the set of configurations of any prime event structure since it is not closed under (unbounded) intersections:

$$\{e_1, e_3\} \cap \{e_2, e_3\} = \{e_3\}$$

but $\{e_3\}$ is not a configuration.

Prime event structures are a proper subclass of stable event structures (which we do not define here). Any stable configuration structure is the set of configurations of a stable event structure (van Glabbeek and Goltz 2001, Theorem 5.3).

Definition 3.6. Let $\mathcal{C} = (C, \ell)$ be a stable configuration structure, and let $X \in C$ with $d, e \in X$. Then we have:

— Causality, that is, $d \leqslant_X e$ if and only if for all $Y \in C$ with $Y \subseteq X$ we have $e \in Y$ implies $d \in Y$. Furthermore, $d <_X e$ if and only if $d \leqslant_X e$ and $d \neq e$.
— Concurrency, that is, $d \; co_X \; e$ if and only if $d \not\leqslant_X e$ and $e \not\leqslant_X d$.

Example 3.7. Consider the stable configuration structure $\mathcal{C}$ of Example 3.5. We have

$$e_1 <_{\{e_1, e_3\}} e_3$$
$$e_2 <_{\{e_2, e_3\}} e_3.$$

Thus e_3 can be caused by either e_1 or e_2, but not both. This is an example of exclusive 'or' causation, which cannot be modelled (directly) in prime event structures.

https://doi.org/10.1017/S0960129513000510 Published online by Cambridge University Press

Note that the causal relations are local to configurations, unlike the case with prime event structures where there is a single global causal ordering. van Glabbeek and Goltz showed (van Glabbeek and Goltz 2001) that $<_X$ is a partial order and that the sub-configurations of X are precisely those subsets Y that are left-closed with respect to $<_X$, that is, if $d <_X e \in Y$, then $d \in Y$. Furthermore, if $X, Y \in C$ with $Y \subseteq X$, then $<_Y = <_X \upharpoonright Y$.

Definition 3.8. Let $\mathcal{C} = (C, \ell)$ be a stable configuration structure and let $a \in \mathsf{Act}$. We let $X \xrightarrow{e}_{\mathcal{C}} X'$ if and only if $X, X' \in C$, $X \subseteq X'$ and $X' \setminus X = \{e\}$. Furthermore, we let $X \xrightarrow{a}_{\mathcal{C}} X'$ if and only if $X \xrightarrow{e}_{\mathcal{C}} X'$ for some e with $\ell(e) = a$. We also define reverse transitions: $X \overset{e}{\rightsquigarrow}_{\mathcal{C}} X'$ if and only if $X' \xrightarrow{e}_{\mathcal{C}} X$, and $X \overset{a}{\rightsquigarrow}_{\mathcal{C}} X'$ if and only if $X' \xrightarrow{a}_{\mathcal{C}} X$. The overloading of notation whereby transitions can be labelled with events or with event labels should not cause confusion.

We shall assume in the following that stable configuration structures are *image finite* with respect to forward transitions, that is, for any configuration X and any label a, the set $\{X' : X \xrightarrow{a}_{\mathcal{C}} X'\}$ is finite.

3.2. *Equivalences*

In this section we define the bisimulation-based equivalences we shall need, namely, those shown in Figure 1, and give examples that demonstrate the differences between them.

Definition 3.9 (van Glabbeek and Goltz 2001). Let $\mathcal{C}$ and $\mathcal{D}$ be stable configuration structures. A relation $\mathcal{R} \subseteq C_{\mathcal{C}} \times C_{\mathcal{D}}$ is an *interleaving bisimulation* (IB) between $\mathcal{C}$ and $\mathcal{D}$ if $\mathcal{R}(\emptyset, \emptyset)$, and if $\mathcal{R}(X, Y)$, then for $a \in \mathsf{Act}$:

— if $X \xrightarrow{a}_{\mathcal{C}} X'$, then $\exists Y'.\ Y \xrightarrow{a}_{\mathcal{D}} Y'$ and $\mathcal{R}(X', Y')$;
— if $Y \xrightarrow{a}_{\mathcal{D}} Y'$, then $\exists X'.\ X \xrightarrow{a}_{\mathcal{C}} X'$ and $\mathcal{R}(X', Y')$.

We say that $\mathcal{C}$ and $\mathcal{D}$ are IB equivalent ($\mathcal{C} \approx_{\mathsf{ib}} \mathcal{D}$) if and only if there is an IB between $\mathcal{C}$ and $\mathcal{D}$.

Example 3.10. Consider a configuration structure $\mathcal{C}$ that has events e_1 and e_2 with labels a and b, respectively, and the configurations $\emptyset$, $\{e_1\}, \{e_2\}$ and $\{e_1, e_2\}$. The corresponding CCS expression is $a \,|\, b$. Clearly, we have

$$\emptyset \xrightarrow{a}_{\mathcal{C}} \{e_1\}$$
$$\emptyset \xrightarrow{b}_{\mathcal{C}} \{e_2\}.$$

Next, consider a configuration structure $\mathcal{D}$ that consists of $\emptyset$, $\{d_1\}$ and $\{d_1, d_2\}$ where the events d_1 and d_2 are labelled a and b, respectively. The corresponding CCS expression is $a.b$, and we have $\emptyset \xrightarrow{a}_{\mathcal{D}} \{d_1\}$ but not $\emptyset \xrightarrow{b}_{\mathcal{D}} Y$ for any configuration Y of $\mathcal{D}$. Hence, $\mathcal{C}$ and $\mathcal{D}$ are not IB equivalent.

For a set of events E, let $\ell(E)$ be the multiset of labels of events in E. We shall now define a *step* transition relation where concurrent events are executed in a single step.

https://doi.org/10.1017/S0960129513000510 Published online by Cambridge University Press

Definition 3.11. Let $\mathcal{C} = (C, \ell)$ be a stable configuration structure and let $A \in \mathbb{N}^{\mathsf{Act}}$ (A is a multiset over Act). We let $X \xrightarrow{A}_{\mathcal{C}} X'$ if and only if $X, X' \in C$, $X \subseteq X'$ and $X' \setminus X = E$ with $d\ co_{X'}\ e$ for all $d, e \in E$ and $\ell(E) = A$.

Example 3.12. Consider the configuration structure $\mathcal{C}$ from Example 3.10. Since e_1 and e_2 are concurrent, we have $\varnothing \xrightarrow{\{a,b\}}_{\mathcal{C}} \{e_1, e_2\}$.

Definition 3.13 (Pomello 1986; van Glabbeek and Goltz 2001). Let $\mathcal{C}$ and $\mathcal{D}$ be stable configuration structures. A relation $\mathcal{R} \subseteq C_{\mathcal{C}} \times C_{\mathcal{D}}$ is a *step bisimulation* (SB) between $\mathcal{C}$ and $\mathcal{D}$ if $\mathcal{R}(\varnothing, \varnothing)$, and if $\mathcal{R}(X, Y)$, then for $A \in \mathbb{N}^{\mathsf{Act}}$:

— if $X \xrightarrow{A}_{\mathcal{C}} X'$, then $\exists Y'.\ Y \xrightarrow{A}_{\mathcal{D}} Y'$ and $\mathcal{R}(X', Y')$;
— if $Y \xrightarrow{A}_{\mathcal{D}} Y'$, then $\exists X'.\ X \xrightarrow{A}_{\mathcal{C}} X'$ and $\mathcal{R}(X', Y')$.

We say that $\mathcal{C}$ and $\mathcal{D}$ are SB equivalent ($\mathcal{C} \approx_{\mathsf{sb}} \mathcal{D}$) if and only if there is an SB between $\mathcal{C}$ and $\mathcal{D}$.

Example 3.14. Consider the two configuration structures from Example 3.10, but now with all labels being a. The corresponding CCS expressions are $a \mid a$ and $a.a$ and they are IB equivalent. However, step bisimulation distinguishes them since

$$\varnothing \xrightarrow{\{a,a\}}_{\mathcal{C}} \{e_1, e_2\}$$

but not

$$\varnothing \xrightarrow{\{a,a\}}_{\mathcal{D}} \{d_1, d_2\}.$$

The last transition does not hold since

$$d_1 <_{\{d_1,d_2\}} d_2.$$

Definition 3.15. Let

$$\mathcal{X} = (X, <_X, \ell_X)$$
$$\mathcal{Y} = (Y, <_Y, \ell_Y)$$

be partial orders that are labelled over Act. We say that $\mathcal{X}$ and $\mathcal{Y}$ are *isomorphic* ($\mathcal{X} \cong \mathcal{Y}$, or sometimes just $X \cong Y$) if and only if there is a bijection from X to Y respecting the ordering and the labelling. The isomorphism class $[\mathcal{X}]_{\cong}$ of a partial order labelled over Act is called a *pomset* over Act. We let $p, \ldots$ range over pomsets.

Thus a pomset is an abstraction of a labelled partial order where we forget about the actual events and just consider it as an ordering on a multiset of labels.

Definition 3.16. Let $\mathcal{C} = (C, \ell)$ be a stable configuration structure and p be a pomset over Act. We let $X \xrightarrow{p}_{\mathcal{C}} X'$ if and only if $X, X' \in C$, $X \subseteq X'$ and $X' \setminus X = H$ with

$$p = [(H, <_{X'} \cap (H \times H), \ell_{\mathcal{C}} \restriction H)]_{\cong}\ .$$

https://doi.org/10.1017/S0960129513000510 Published online by Cambridge University Press

Example 3.17. Consider $\mathcal{C}$ and $\mathcal{D}$ in Example 3.10. Let $a < b$ denote the pomset of a partial order consisting of two events labelled a and b where the first event causes the second event. Then

$$\emptyset \stackrel{\{a<b\}}{\rightarrow}_{\mathcal{D}} \{d_1, d_2\}$$

since

$$d_1 <_{\{d_1,d_2\}} d_2.$$

However, it is not true that

$$\emptyset \stackrel{\{a<b\}}{\rightarrow}_{\mathcal{C}} \{e_1, e_2\},$$

since e_1 and e_2 are concurrent.

Definition 3.18 (Boudol and Castellani 1987; van Glabbeek and Goltz 2001). Let $\mathcal{C}$ and $\mathcal{D}$ be stable configuration structures. A relation $\mathcal{R} \subseteq C_{\mathcal{C}} \times C_{\mathcal{D}}$ is a *pomset bisimulation* (PB) between $\mathcal{C}$ and $\mathcal{D}$ if $\mathcal{R}(\emptyset, \emptyset)$, and if $\mathcal{R}(X, Y)$, then for any pomset p over Act:

— if $X \stackrel{p}{\rightarrow}_{\mathcal{C}} X'$, then $\exists Y'.\ Y \stackrel{p}{\rightarrow}_{\mathcal{D}} Y'$ and $\mathcal{R}(X', Y')$;
— if $Y \stackrel{p}{\rightarrow}_{\mathcal{D}} Y'$, then $\exists X'.\ X \stackrel{p}{\rightarrow}_{\mathcal{C}} X'$ and $\mathcal{R}(X', Y')$.

We say that $\mathcal{C}$ and $\mathcal{D}$ are PB equivalent ($\mathcal{C} \approx_{\mathsf{pb}} \mathcal{D}$) if and only if there is a PB between $\mathcal{C}$ and $\mathcal{D}$.

Example 3.19. Consider the configuration structures $\mathcal{C}$ and $\mathcal{D}$ corresponding to the CCS expressions $(a \,|\, a) + a.a$ and $a \,|\, a$. We have that $\mathcal{C}$ can perform the pomset $a < a$ but $\mathcal{D}$ cannot; hence they are not PB equivalent. Note, however, that $\mathcal{C}$ and $\mathcal{D}$ are SB equivalent.

Definition 3.20 (Degano *et al.* 1987; van Glabbeek and Goltz 2001). Let $\mathcal{C}$ and $\mathcal{D}$ be stable configuration structures. A relation $\mathcal{R} \subseteq C_{\mathcal{C}} \times C_{\mathcal{D}}$ is a *weak history-preserving (WH) bisimulation* between $\mathcal{C}$ and $\mathcal{D}$ if $\mathcal{R}(\emptyset, \emptyset)$, and if $\mathcal{R}(X, Y)$ and $a \in \mathsf{Act}$, then:

— $(X, <_X, \ell_{\mathcal{C}} \restriction X) \cong (Y, <_Y, \ell_{\mathcal{D}} \restriction Y)$;
— if $X \stackrel{a}{\rightarrow}_{\mathcal{C}} X'$, then $\exists Y'.\ Y \stackrel{a}{\rightarrow}_{\mathcal{D}} Y'$ and $\mathcal{R}(X', Y')$;
— if $Y \stackrel{a}{\rightarrow}_{\mathcal{D}} Y'$, then $\exists X'.\ X \stackrel{a}{\rightarrow}_{\mathcal{C}} X'$ and $\mathcal{R}(X', Y')$.

We say that $\mathcal{C}$ and $\mathcal{D}$ are WH equivalent ($\mathcal{C} \approx_{\mathsf{wh}} \mathcal{D}$) if and only if there is a WH bisimulation between $\mathcal{C}$ and $\mathcal{D}$.

Example 3.21. Consider the configuration structures $\mathcal{C}$ and $\mathcal{D}$ corresponding to the CCS expressions $a.(b + c) + (a \,|\, b) + a.b$ and $a.(b + c) + (a \,|\, b)$. They are PB equivalent because the a of $a.b$ in $\mathcal{C}$ is matched by the a of $a \,|\, b$ in $\mathcal{D}$ and then can be followed by matching bs. This does not work for WH bisimulation because after the said a and b in $\mathcal{C}$ we are in a configuration where b depends causally on a, and after the matching a and b in $\mathcal{D}$ we reach a configuration where a and b are concurrent. This violates the property of WH bisimulation that matching configurations are order-isomorphic.

We can define a further equivalence by combining pomset and weak-history preserving bisimulation as follows.

Definition 3.22 (van Glabbeek and Goltz 2001). Let $\mathcal{C}$ and $\mathcal{D}$ be stable configuration structures. A relation $\mathcal{R} \subseteq C_{\mathcal{C}} \times C_{\mathcal{D}}$ is a *weak history-preserving pomset bisimulation (WHPB)* between $\mathcal{C}$ and $\mathcal{D}$ if $\mathcal{R}(\emptyset, \emptyset)$, and if $\mathcal{R}(X, Y)$ and p is a pomset over Act, then:

— $(X, <_X, \ell_{\mathcal{C}} \upharpoonright X) \cong (Y, <_Y, \ell_{\mathcal{D}} \upharpoonright Y)$;
— if $X \xrightarrow{p}_{\mathcal{C}} X'$, then $\exists Y'.\ Y \xrightarrow{p}_{\mathcal{D}} Y'$ and $\mathcal{R}(X', Y')$;
— if $Y \xrightarrow{p}_{\mathcal{D}} Y'$, then $\exists X'.\ X \xrightarrow{p}_{\mathcal{C}} X'$ and $\mathcal{R}(X', Y')$.

We say that $\mathcal{C}$ and $\mathcal{D}$ are WHPB equivalent ($\mathcal{C} \approx_{\mathsf{whpb}} \mathcal{D}$) if and only if there is a WHPB between $\mathcal{C}$ and $\mathcal{D}$.

Definition 3.23 (Rabinovich and Trakhtenbrot 1988; van Glabbeek and Goltz 2001). Let $\mathcal{C}$ and $\mathcal{D}$ be stable configuration structures. A relation

$$\mathcal{R} \subseteq C_{\mathcal{C}} \times C_{\mathcal{D}} \times \mathcal{P}(E_{\mathcal{C}} \times E_{\mathcal{D}})$$

is a *history-preserving (H) bisimulation* between $\mathcal{C}$ and $\mathcal{D}$ if and only if $\mathcal{R}(\emptyset, \emptyset, \emptyset)$, and if $\mathcal{R}(X, Y, f)$ and $a \in \mathsf{Act}$:

— f is an isomorphism between $(X, <_X, \ell_{\mathcal{C}} \upharpoonright X)$ and $(Y, <_Y, \ell_{\mathcal{D}} \upharpoonright Y)$;
— if $X \xrightarrow{a}_{\mathcal{C}} X'$, then $\exists Y', f'.\ Y \xrightarrow{a}_{\mathcal{D}} Y'$, $\mathcal{R}(X', Y', f')$ and $f' \upharpoonright X = f$;
— if $Y \xrightarrow{a}_{\mathcal{D}} Y'$, then $\exists X', f'.\ X \xrightarrow{a}_{\mathcal{C}} X'$, $\mathcal{R}(X', Y', f')$ and $f' \upharpoonright X = f$.

We say that $\mathcal{C}$ and $\mathcal{D}$ are H equivalent ($\mathcal{C} \approx_{\mathsf{h}} \mathcal{D}$) if and only if there is an H bisimulation between $\mathcal{C}$ and $\mathcal{D}$.

Both H and WH have associated hereditary versions as follows.

Definition 3.24 (Bednarczyk 1991; Joyal *et al.* 1996; van Glabbeek and Goltz 2001). Let $\mathcal{C}$ and $\mathcal{D}$ be stable configuration structures. Then

$$\mathcal{R} \subseteq C_{\mathcal{C}} \times C_{\mathcal{D}} \times \mathcal{P}(E_{\mathcal{C}} \times E_{\mathcal{D}})$$

is a *hereditary H (HH) bisimulation* if and only if $\mathcal{R}$ is an H bisimulation with the additional *hereditary* property that if $\mathcal{R}(X, Y, f)$, then for any $a \in \mathsf{Act}$:

— if $X \overset{a}{\rightsquigarrow}_{\mathcal{C}} X'$, then $\exists Y', f'.\ Y \overset{a}{\rightsquigarrow}_{\mathcal{D}} Y'$, $\mathcal{R}(X', Y', f')$ and $f \upharpoonright X' = f'$;
— if $Y \overset{a}{\rightsquigarrow}_{\mathcal{D}} Y'$, then $\exists X', f'.\ X \overset{a}{\rightsquigarrow}_{\mathcal{C}} X'$, $\mathcal{R}(X', Y', f')$ and $f \upharpoonright X' = f'$.

We say that $\mathcal{C}$ and $\mathcal{D}$ are HH equivalent ($\mathcal{C} \approx_{\mathsf{hh}} \mathcal{D}$) if and only if there is an HH bisimulation between $\mathcal{C}$ and $\mathcal{D}$.

Definition 3.25. Let $\mathcal{C}$ and $\mathcal{D}$ be stable configuration structures. Then

$$\mathcal{R} \subseteq C_{\mathcal{C}} \times C_{\mathcal{D}} \times \mathcal{P}(E_{\mathcal{C}} \times E_{\mathcal{D}})$$

is a *hereditary WH (HWH) bisimulation* if $\mathcal{R}(\emptyset, \emptyset, \emptyset)$, and if $\mathcal{R}(X, Y, f)$ and $a \in \mathsf{Act}$, then:

— f is an isomorphism between $(X, <_X, \ell_{\mathcal{C}} \upharpoonright X)$ and $(Y, <_Y, \ell_{\mathcal{D}} \upharpoonright Y)$;
— if $X \xrightarrow{a}_{\mathcal{C}} X'$, then $\exists Y', f'.\ Y \xrightarrow{a}_{\mathcal{D}} Y'$ and $\mathcal{R}(X', Y', f')$;
— if $Y \xrightarrow{a}_{\mathcal{D}} Y'$, then $\exists X', f'.\ X \xrightarrow{a}_{\mathcal{C}} X'$ and $\mathcal{R}(X', Y', f')$;
— if $X \overset{a}{\rightsquigarrow}_{\mathcal{C}} X'$, then $\exists Y', f'.\ Y \overset{a}{\rightsquigarrow}_{\mathcal{D}} Y'$, $\mathcal{R}(X', Y', f')$ and $f \upharpoonright X' = f'$;

https://doi.org/10.1017/S0960129513000510 Published online by Cambridge University Press

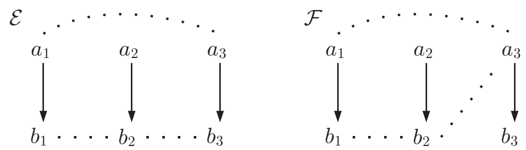

Fig. 2. $\mathcal{E} \approx_{\mathsf{hwh}} \mathcal{F}$, but $\mathcal{E} \not\approx_{\mathsf{pb}} \mathcal{F}$.

— if $Y \overset{a}{\rightsquigarrow}_{\mathcal{D}} Y'$, then $\exists X', f'.\ X \overset{a}{\rightsquigarrow}_{\mathcal{C}} X'$, $\mathcal{R}(X', Y', f')$ and $f \upharpoonright X' = f'$.

Also, $\mathcal{C}$ and $\mathcal{D}$ are HWH equivalent ($\mathcal{C} \approx_{\mathsf{hwh}} \mathcal{D}$) if and only if there is an HWH bisimulation between $\mathcal{C}$ and $\mathcal{D}$.

To see that if $\mathcal{C} \approx_{\mathsf{hwh}} \mathcal{D}$, then $\mathcal{C} \approx_{\mathsf{wh}} \mathcal{D}$, we suppose that $\mathcal{R}(X, Y, f)$ is an HWH bisimulation between $\mathcal{C}$ and $\mathcal{D}$. We define $\mathcal{R}'(X, Y)$ if and only if $\exists f.\mathcal{R}(X, Y, f)$ and can easily check that $\mathcal{R}'$ is a WH bisimulation between $\mathcal{C}$ and $\mathcal{D}$.

The other inclusions in Figure 1 are mostly immediate from the definitions. The inclusion $\approx_{\mathsf{wh}} \subseteq \approx_{\mathsf{sb}}$ is non-obvious; it was shown in Fecher (2004), with an alternative proof given in Phillips and Ulidowski (2012). Furthermore, the inclusions in Figure 1 are all strict, and no further inclusions hold between the specified equivalences, as we now show by means of a series of six examples collected together in Example 3.26.

Example 3.26.

(1) Phillips and Ulidowski (2012, Example 3.12):

$$a \mid a = a.a$$

holds for IB but not SB, as explained in Example 3.14.

(2) Phillips and Ulidowski (2012, Example 3.13):

$$a \mid a = (a \mid a) + a.a$$

holds for SB but not PB (see Example 3.19) or WH (it is clear that $(a \mid a) + a.a$ can reach a configuration corresponding to $a.a$ that is not order-isomorphic to any configuration of $a \mid a$).

(3) van Glabbeek and Goltz (2001, Example 9.1) and Phillips and Ulidowski (2012, Example 3.14):

$$a.(b + c) + (a \mid b) + a.b = a.(b + c) + (a \mid b)$$

holds for PB but not WH, as shown in Example 3.21.

(4) Phillips and Ulidowski (2012, Example 4.7):

The event structures $\mathcal{E}$, $\mathcal{F}$ in Figure 2 are HWH-equivalent but not PB-equivalent. Recall that here, as elsewhere, when we label events as $a_1, a_2, \ldots$, we mean that there are distinct events $e_1, e_2, \ldots$ that are labelled with a. We shall first show that $\mathcal{E}$ and $\mathcal{F}$ are not PB-equivalent. In $\mathcal{E}$, after any a, we can always perform a pomset transition $a < b$, whereas in $\mathcal{F}$, we cannot perform $a < b$ after a_3 (though a_2 and b_3 are possible). Next, we check that $\mathcal{E}$, $\mathcal{F}$ are HWH-equivalent. Note that every configuration of $\mathcal{F}$ has a corresponding configuration in $\mathcal{E}$. The only difference is that configuration $\{a_2, b_2, a_3\}$ of $\mathcal{E}$ is missing in $\mathcal{F}$. This configuration is matched by $\{a_2, b_2, a_1\}$ and by

https://doi.org/10.1017/S0960129513000510 Published online by Cambridge University Press

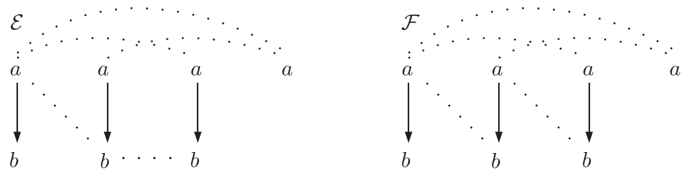

Fig. 3. $\mathcal{E} \approx_{\mathsf{whpb}} \mathcal{F}$, but $\mathcal{E} \not\approx_{\mathsf{h}} \mathcal{F}$ and $\mathcal{E} \not\approx_{\mathsf{hwh}} \mathcal{F}$.

$\{a_2, a_3, b_3\}$ in $\mathcal{F}$. We now define a relation R between the matching configurations of $\mathcal{E}$ and $\mathcal{F}$ and check that it is an HWH bisimulation. Crucially, we check that there are order isomorphisms between $\{a_2, b_2, a_3\}$ of $\mathcal{E}$ and its matching configurations in $\mathcal{F}$, namely

$$\{(a_2, a_2), (b_2, b_2), (a_3, a_1)\}$$

and

$$\{(a_2, a_3), (b_2, b_3), (a_3, a_2)\}.$$

Finally, we see that reversing any pair of isomorphic events (that can be reversed) leads to related configurations.

(5) van Glabbeek and Goltz (2001, Example 9.4, page 294) and Phillips and Ulidowski (2012, Example 3.16):
The event structures $\mathcal{E}$, $\mathcal{F}$ in Figure 3 are WHPB-equivalent, but not H-equivalent (nor in fact HWH-equivalent, though this is not needed for our current purposes). We shall first show that $\mathcal{E}$ and $\mathcal{F}$ are not H-equivalent. Consider the two middle events a in $\mathcal{E}$, denoted here by a and a' as read from the left, and configuration $E_{aa'}$ consisting of these events. $E_{aa'}$ can be extended to $E_{aa'b}$ by performing the middle b, and to $E_{aa'b'}$ by performing the b on the right, which is denoted here by b'. There are three configurations in $\mathcal{F}$ consisting of two as, and each of these configurations can be extended by performing a single event b. We write these configurations as $F_{aa'}$ and $F_{aa'b}$. Next we check that $F_{aa'}$ cannot be related to $E_{aa'}$ by an H bisimulation. This is because after fixing which a, a' in $E_{aa'}$ match which a, a' in $F_{aa'}$, we cannot ensure that both $E_{aa'b}$ and $E_{aa'b'}$ are order-isomorphic to $F_{aa'b}$, assuming that we maintain the matching between the events of $E_{aa'}$ and $F_{aa'}$.
However, this is not a problem when defining a WH bisimulation because there is no requirement that the events matched so far must stay matched in future configurations. Each configuration $F_{aa'b}$ is related to $E_{aa'b}$ and $E_{aa'b'}$ because we are allowed to redefine which a, a' in $E_{aa'}$ matches which a, a' in $F_{aa'}$. Hence, $\mathcal{E}$, $\mathcal{F}$ are WH-equivalent, and since they also have matching pomsets, they are WHPB-equivalent.

(6) The Absorption Law (Boudol and Castellani 1987; Bednarczyk 1991; van Glabbeek and Goltz 2001):

$$(a \,|\, (b + c)) + (a \,|\, b) + ((a + c) \,|\, b) = (a \,|\, (b + c)) + ((a + c) \,|\, b)$$

holds for H, and thus for WH, but not for HWH, which we shall now demonstrate by showing that HWH bisimulation distinguishes the two sides of the Absorption Law.

https://doi.org/10.1017/S0960129513000510 Published online by Cambridge University Press

If we perform the event a and then b in the $a \mid b$ component, these must be matched by the a and then the b of the $((a+c) \mid b)$ summand on the right (matching it with the a of $(a \mid (b+c))$ is wrong since after this a is performed, no c is possible after a in $a \mid b$). The right-hand side can now reverse a and do a c (still using the same summand since all other summands are disabled). The left-hand side cannot match this since, after reversing the a, no c is possible.

4. Event Identifier Logic

We now introduce our logic, which we call Event Identifier Logic (EIL). We assume an infinite set of identifiers Id ranged over by $x, y, z, \ldots$. The syntax of EIL is as follows:

$$\phi ::= \mathrm{tt} \mid \neg\phi \mid \phi \wedge \phi' \mid \langle x : a \rangle\rangle \phi \mid (x : a)\phi \mid \langle\langle x \rangle \phi.$$

We include the usual operators of propositional logic: truth tt, negation $\neg\phi$ and conjunction $\phi \wedge \phi'$. We then have *forward diamond* $\langle x : a \rangle\rangle \phi$, which says that it is possible to perform an event labelled with a and reach a new configuration where ϕ holds. In the formula $\langle x : a \rangle\rangle \phi$, the modality $\langle x : a \rangle\rangle$ binds all free occurrences of x in ϕ. Next we have *declaration* $(x : a)\phi$, which says that there is some event with label a in the current configuration that can be bound to x in such a way that ϕ holds. Here the declaration $(x : a)$ binds all free occurrences of x in ϕ. Finally, we have *reverse diamond* $\langle\langle x \rangle \phi$, which says that it is possible to perform the reverse event bound to identifier x, and reach a configuration where ϕ holds. Note that $\langle\langle x \rangle$ does not bind x. It is clear that any occurrences of x that get bound by $(x : a)$ must be of the form $\langle\langle x \rangle$. We allow alpha-conversion of bound names. We use $\phi, \psi, \ldots$ to range over formulas of EIL.

Example 4.1. The formula

$$\langle x : a \rangle\rangle \langle y : a \rangle\rangle \langle\langle x \rangle \mathrm{tt}$$

says that there are events with label a, say e_1 and e_2, that can be bound to x and y such that, after performing e_1 and then e_2, we can reverse e_1. Obviously, after performing e_1 followed by e_2, we can always reverse e_2. This formula could be interpreted as saying that an event bound to x is *concurrent* with (or *independent* of) an event bound to y. Next, consider

$$\langle x : a \rangle\rangle \langle y : a \rangle\rangle \neg \langle\langle x \rangle \mathrm{tt}.$$

This formula expresses the fact that an event bound to x *causes* an event bound to y (because if we could reverse x before y, we would reach a configuration containing y and not x, which contradicts x being a cause of y).

Definition 4.2. We define $\mathsf{fi}(\phi)$, the set of free identifiers of ϕ, by induction on formulas:

$$\begin{aligned} \mathsf{fi}(\mathsf{tt}) &= \varnothing \\ \mathsf{fi}(\neg\phi) &= \mathsf{fi}(\phi) \\ \mathsf{fi}(\phi_1 \wedge \phi_2) &= \mathsf{fi}(\phi_1) \cup \mathsf{fi}(\phi_2) \\ \mathsf{fi}(\langle x : a\rangle\rangle\phi) &= \mathsf{fi}(\phi) \setminus \{x\} \\ \mathsf{fi}((x : a)\phi) &= \mathsf{fi}(\phi) \setminus \{x\} \\ \mathsf{fi}(\langle\langle x\rangle\phi) &= \mathsf{fi}(\phi) \cup \{x\}. \end{aligned}$$

We say that ϕ is *closed* if $\mathsf{fi}(\phi) = \varnothing$; otherwise ϕ is *open*.

As usual, in order to assign meaning to open formulas, we employ environments that tell us what events the free identifiers are bound to.

Definition 4.3. An *environment* ρ is a partial mapping from Id to events. We say that ρ *is a permissible environment for* ϕ *and* X if

$$\mathsf{fi}(\phi) \subseteq \mathsf{dom}(\rho)$$

and

$$\mathsf{rge}(\rho \restriction \mathsf{fi}(\phi)) \subseteq X.$$

We shall use ρ_ϕ as an abbreviation for $\rho \restriction \mathsf{fi}(\phi)$, so the latter condition can be written as

$$\mathsf{rge}(\rho_\phi) \subseteq X.$$

We use:

— $\varnothing$ to denote the empty environment;
— $\rho[x \mapsto e]$ to denote the environment ρ' that agrees with ρ except possibly on x, where $\rho'(x) = e$ (and $\rho(x)$ may or may not be defined);
— $[x \mapsto e]$ as an abbreviation for $\varnothing[x \mapsto e]$;
— $\rho \setminus x$ to denote ρ with the assignment to x deleted (if defined in ρ).

We can now formally define the semantics of EIL.

Definition 4.4. Let $\mathcal{C}$ be a stable configuration structure. We define a satisfaction relation $\mathcal{C}, X, \rho \models \phi$, where X is a configuration of $\mathcal{C}$ and ρ is a permissible environment for ϕ and X, by induction on formulas as follows (we suppress the $\mathcal{C}$ where it is clear from the context):

— $X, \rho \models \mathsf{tt}$ always
— $X, \rho \models \neg\phi$ if and only if $X, \rho \not\models \phi$
— $X, \rho \models \phi_1 \wedge \phi_2$ if and only if $X, \rho \models \phi_1$ and $X, \rho \models \phi_2$
— $X, \rho \models \langle x : a\rangle\rangle\phi$ if and only if $\exists X', e$ such that we have $X \xrightarrow{e}_{\mathcal{C}} X'$ with $\ell(e) = a$ and $X', \rho[x \mapsto e] \models \phi$
— $X, \rho \models (x : a)\phi$ if and only if $\exists e \in X$ such that $\ell(e) = a$ and $X, \rho[x \mapsto e] \models \phi$
— $X, \rho \models \langle\langle x\rangle\phi$ if and only if $\exists X', e$ such that $X \overset{e}{\rightsquigarrow}_{\mathcal{C}} X'$ with $\rho(x) = e$ and $X', \rho \models \phi$ (and ρ is a permissible environment for ϕ and X').

https://doi.org/10.1017/S0960129513000510 Published online by Cambridge University Press

For closed ϕ, we further define $\mathcal{C}, X \models \phi$ if and only if $\mathcal{C}, X, \varnothing \models \phi$, and $\mathcal{C} \models \phi$ if and only if $\mathcal{C}, \varnothing \models \phi$.

Note that in the case of $\langle\langle x \rangle \phi$, even though according to the syntax x is allowed to occur free in ϕ, if x does occur free in ϕ, then $X, \rho \models \langle\langle x \rangle \phi$ can never hold: if $\rho(x) = e$ and $X \overset{e}{\rightsquigarrow}_{\mathcal{C}} X'$, then $X', \rho \models \phi$ cannot hold since ρ is not a permissible environment for ϕ and X' since ρ assigns a free identifier of ϕ to an event outside X'.

Example 4.5. Consider the configuration structure from Example 3.2. Recall that this has events e_1, e_2, e_3 all labelled with a, where e_1 causes e_2 and e_1, e_2 are concurrent with e_3. The corresponding CCS expression is $(a.a)|a$ and the configurations are

$$\varnothing \quad \{e_1\} \quad \{e_3\} \quad \{e_1, e_2\} \quad \{e_1, e_3\} \quad \{e_1, e_2, e_3\}.$$

To see that the empty configuration satisfies $\langle x : a \rangle\rangle \langle y : a \rangle\rangle \langle\langle x \rangle \mathrm{tt}$, we have

$$\varnothing, \varnothing \models \langle x : a \rangle\rangle \langle y : a \rangle\rangle \langle\langle x \rangle \mathrm{tt}$$

since

$$\{e_1, e_3\}, [x \mapsto e_1, y \mapsto e_3] \models \langle\langle x \rangle \mathrm{tt},$$

which holds because $\{e_1, e_3\} \overset{e_1}{\rightsquigarrow} \{e_3\}$ and $\rho(x) = e_1$.
Also,

$$\varnothing, \varnothing \models \langle x : a \rangle\rangle \langle y : a \rangle\rangle \neg \langle\langle x \rangle \mathrm{tt}$$

since

$$\{e_1, e_2\}, [x \mapsto e_1, y \mapsto e_2] \models \neg \langle\langle x \rangle \mathrm{tt},$$

because

$$\{e_1, e_2\} \overset{e_1}{\not\rightsquigarrow} \{e_2\}$$

since $\{e_2\}$ is not a configuration.

The closed formula $(x : a)\mathrm{tt}$ says that there is some event labelled with a in the current configuration: $X \models (x : a)\mathrm{tt}$ if and only if $\exists e \in X.\ \ell(e) = a$. In the present example, note that as well as

$$\{e_1, e_2\}, [x \mapsto e_1, y \mapsto e_2] \models \neg \langle\langle x \rangle \mathrm{tt}$$

we also have

$$\{e_1, e_2\}, [x \mapsto e_1, y \mapsto e_2] \models (x : a) \langle\langle x \rangle \mathrm{tt}.$$

By the definition of $(x : a)$, the current environment is updated to

$$[x \mapsto e_2, y \mapsto e_2]$$

and we obtain

$$\{e_1, e_2\}, [x \mapsto e_2, y \mapsto e_2] \models \langle\langle x \rangle \mathrm{tt}.$$

Correspondingly,

$$\{e_1, e_2\}, [x \mapsto e_1, y \mapsto e_2] \models (x : a) \langle\langle x \rangle (y : a) \langle\langle y \rangle \mathrm{tt}.$$

https://doi.org/10.1017/S0960129513000510 Published online by Cambridge University Press

However,

$$\{e_1, e_2\}, [x \mapsto e_1, y \mapsto e_2] \not\models (x : a)\langle\!\langle x \rangle\langle\!\langle y \rangle \mathtt{tt}$$

since

$$\{e_1\}, [x \mapsto e_2, y \mapsto e_2] \not\models \langle\!\langle y \rangle \mathtt{tt}.$$

We will now introduce some further operators as derived operators of EIL.

Notation 4.6 (derived operators). Let $A = \{a_1, \ldots, a_n\}$ be a multiset of labels. We define:

$$\mathtt{ff} \stackrel{\mathrm{df}}{=} \neg \mathtt{tt}$$

$$[x : a]]\, \phi \stackrel{\mathrm{df}}{=} \neg \langle x : a \rangle\!\rangle \neg \phi$$

$$\phi_1 \vee \phi_2 \stackrel{\mathrm{df}}{=} \neg(\neg \phi_1 \wedge \neg \phi_2)$$

(*Forward step*) $$\langle A \rangle\!\rangle \phi \stackrel{\mathrm{df}}{=} \langle x_1 : a_1 \rangle\!\rangle \cdots \langle x_n : a_n \rangle\!\rangle \left(\phi \wedge \bigwedge_{i=1}^{n-1} \langle\!\langle x_i \rangle \mathtt{tt} \right)$$

where $x_1, \ldots, x_n$ are fresh and distinct (and, in particular, are not free in ϕ).
We write $\langle a_1, \ldots, a_n \rangle\!\rangle \phi$ instead of $\langle \{a_1, \ldots, a_n\} \rangle\!\rangle \phi$.
In the case $n = 1$, we have

$$\langle a \rangle\!\rangle \phi \stackrel{\mathrm{df}}{=} \langle x : a \rangle\!\rangle \phi$$

where x is fresh.

(*Reverse step*) $$\langle\!\langle A \rangle \phi \stackrel{\mathrm{df}}{=} (x_1 : a_1) \cdots (x_n : a_n) \left(\langle\!\langle x_1 \rangle \cdots \langle\!\langle x_n \rangle \phi \wedge \bigwedge_{i=2}^{n} \langle\!\langle x_i \rangle \mathtt{tt} \right)$$

where $x_1, \ldots, x_n$ are fresh and distinct (and, in particular, are not free in ϕ).
We write $\langle\!\langle a_1, \ldots, a_n \rangle \phi$ instead of $\langle\!\langle \{a_1, \ldots, a_n\} \rangle \phi$.
In the case $n = 1$, we have

$$\langle\!\langle a \rangle \phi \stackrel{\mathrm{df}}{=} (x : a) \langle\!\langle x \rangle \phi$$

where x is fresh.

The next example gives formulas that distinguish the six pairs of configuration structures in Example 3.26.

Example 4.7.

(1) $\langle x : a \rangle\!\rangle \langle y : a \rangle\!\rangle \langle\!\langle x \rangle \mathtt{tt}$ is satisfied by $a \,|\, a$, but not by $a.a$.
(2) $[x : a]]\ [y : a]]\ \langle\!\langle x \rangle \mathtt{tt}$ is only satisfied by $a \,|\, a$, and not by $(a \,|\, a) + a.a$.
(3) Only the right-hand side of

$$a.(b + c) + (a \,|\, b) = a.(b + c) + (a \,|\, b) + a.b,$$

https://doi.org/10.1017/S0960129513000510 Published online by Cambridge University Press

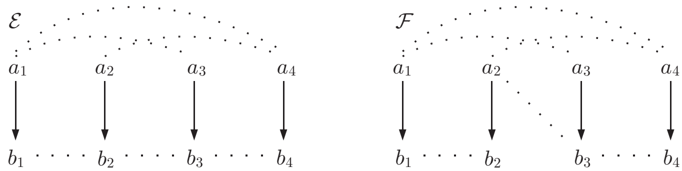

Fig. 4. $\mathcal{E} \approx_{\mathsf{hwh}} \mathcal{F}$ and $\mathcal{E} \approx_{\mathsf{whpb}} \mathcal{F}$ but $\mathcal{E} \not\approx_{\mathsf{h}} \mathcal{F}$.

satisfies

$$\langle a\rangle\rangle([c]]\, \mathrm{ff} \wedge \langle b\rangle\rangle\ [[a]\, \mathrm{ff}).$$

(4) Only $\mathcal{E}$ satisfies

$$[x : a]]\ \langle y : a\rangle\rangle\langle z : b\rangle\rangle\ [[y]\, \mathrm{ff},$$

which says that after every a a pomset $a < b$ can be performed.

(5) Consider

$$\langle x : a\rangle\rangle\langle y : a\rangle\rangle(\langle z : b\rangle\rangle\ [[x]\, \mathrm{ff} \wedge \langle w : b\rangle\rangle\ [[y]\, \mathrm{ff}).$$

This is only satisfied by $\mathcal{E}$ in Figure 3 where the two middle events a are assigned to x and y.

(6) Consider

$$\langle x : a\rangle\rangle([w : c]]\, \mathrm{ff} \wedge \langle y : b\rangle\rangle\langle\langle x\rangle\ [z : c]]\, \mathrm{ff}).$$

This is satisfied by

$$(a\,|\,(b + c)) + (a\,|\,b) + ((a + c)\,|\,b),$$

but not by

$$(a\,|\,(b + c)) + ((a + c)\,|\,b).$$

Strictly speaking, event identifiers are not required to distinguish the two pairs of configuration structures. The formula

$$\langle a\rangle\rangle([c]]\, \mathrm{ff} \wedge \langle b\rangle\rangle\langle\langle a\rangle\ [c]]\, \mathrm{ff})$$

with simple label modalities is sufficient.

Example 4.8. The event structures $\mathcal{E}$ and $\mathcal{F}$ in Figure 4 (which is taken from Phillips and Ulidowski (2012, Example 4.8)) are equivalent for HWH and WHPB, but not for H, and hence not for HH. Now consider

$$\phi \equiv [x : a]]\ [y : a]]\ (\langle z : b\rangle\rangle\neg\langle\langle x\rangle \mathrm{tt} \wedge \langle w : b\rangle\rangle\neg\langle\langle y\rangle \mathrm{tt}).$$

It is easy to check that $\mathcal{E}$ satisfies ϕ and $\mathcal{F}$ does not. Also note that $\mathcal{E}$ and $\mathcal{F}$ can be distinguished by a logic with pomset modalities (both reverse and forward) defined over runs (Cherief 1992; Pinchinat *et al.* 1994).

Example 4.9. Consider $\mathcal{E}$, $\mathcal{F}$ and their configuration graphs in Figure 5. To see that $\mathcal{E}$ and $\mathcal{F}$ are H equivalent, we define a bisimulation by relating configurations of identically labelled events (including where a_4 is matched with a_4') and check that it is an H. The structures are also HWH equivalent. To see this, we define a bisimulation between order-isomorphic configurations (of which there only five isomorphism classes: $\emptyset$, $\{a\}$,

https://doi.org/10.1017/S0960129513000510 Published online by Cambridge University Press

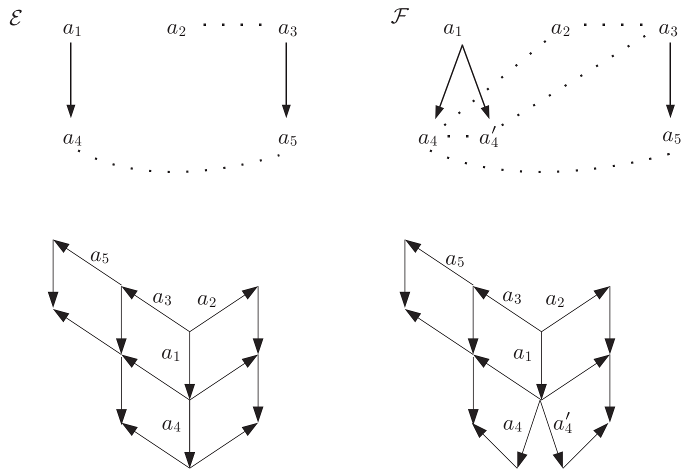

Fig. 5. $\mathcal{E} \approx_{\mathsf{h}} \mathcal{F}$ and $\mathcal{E} \approx_{\mathsf{hwh}} \mathcal{F}$ but $\mathcal{E} \not\approx_{\mathsf{hh}} \mathcal{F}$.

$\{a, a\}$, $\{a < a\}$ and $\{a < a, a\}$, where events separated by commas are concurrent) and check that it is an HWH. However, $\mathcal{E}$ and $\mathcal{F}$ are not HH equivalent, and, since there is autoconcurrency, event identifiers are indeed required to distinguish them. The formula

$$\langle x : a\rangle\rangle\langle y : a\rangle\rangle(\neg\langle\langle x\rangle\mathrm{tt} \wedge \langle z : a\rangle\rangle\langle\langle y\rangle\langle w : a\rangle\rangle\neg\langle\langle z\rangle\mathrm{tt} \wedge \langle z' : a\rangle\rangle\langle\langle y\rangle\neg\langle w' : a\rangle\rangle\neg\langle\langle z'\rangle\mathrm{tt})$$

is only satisfied by $\mathcal{E}$. It requires that x causes y and that z and z' are bound to different events because $\langle z : a\rangle\rangle$ and $\langle z' : a\rangle\rangle$ are followed by mutually contradictory behaviours. This is possible in $\mathcal{E}$ (because a_1, a_4 can be followed by either a_3 or a_2) but not in $\mathcal{F}$ since none of the pairs of causally dependent events offers two different a-events.

Example 4.10. Our logic can characterise (up to isomorphism) the causality and concurrency relationships between events of any configuration. Given any configuration X, we can write a formula θ_X that gives that order structure of X. In fact, θ_X only uses reverse modalities: see Lemma 5.4.

We conclude this section with a basic lemma, which will be useful in Section 5.

Lemma 4.11. Let X be a configuration of a stable configuration structure $\mathcal{C}$, and let $\phi \in$ EIL. Suppose ρ and ρ' are permissible environments for ϕ and X that agree on $\mathsf{fi}(\phi)$. Then $X, \rho \models \phi$ if and only if $X, \rho' \models \phi$.

Proof. The proof is a standard induction on formulas. □

5. Using EIL to characterise equivalences

We wish to show that EIL and its various sublogics characterise the equivalences defined in Section 3.2. Each sublogic of EIL induces an equivalence on configuration structures in a standard fashion.

Definition 5.1. Let L be any sublogic of EIL. Then L induces an equivalence on stable configuration structures as follows: $\mathcal{C} \sim_L \mathcal{D}$ if and only if for all closed $\phi \in L$ we have $\mathcal{C} \models \phi$ if and only if $\mathcal{D} \models \phi$.

In Section 5.1, we shall introduce a simple sublogic that allows us to characterise order isomorphism. Then in Section 5.2, we shall characterise history-preserving equivalences, and in Section 5.3 we shall do the same for pomset and step bisimulation.

5.1. *Reverse-only logic and order isomorphism*

In this section we shall define sublogics of EIL consisting of formulas where only reverse transitions are allowed.

Definition 5.2. *Reverse-only* logic $\mathrm{EIL_{ro}}$ is defined by

$$\phi ::= \mathrm{tt} \mid \neg\phi \mid \phi \wedge \phi' \mid (x : a)\phi \mid \langle\langle x \rangle\rangle \phi.$$

We then define *declaration-free* reverse-only logic $\mathrm{EIL_{dfro}}$ by

$$\phi ::= \mathrm{tt} \mid \neg\phi \mid \phi \wedge \phi' \mid \langle\langle x \rangle\rangle \phi.$$

These logics are preserved between isomorphic configurations, and characterise configurations up to isomorphism, as we shall now show.

Lemma 5.3. Let $\mathcal{C}$ and $\mathcal{D}$ be stable configuration structures, and let X and Y be configurations of $\mathcal{C}$ and $\mathcal{D}$, respectively. Suppose $f : X \cong Y$. Then for any $\phi \in \mathrm{EIL_{ro}}$ and any ρ (a permissible environment for ϕ and X), we have $X, \rho \models \phi$ if and only if $Y, f \circ \rho_\phi \models \phi$.

Proof. We use induction on ϕ. Recall that ρ_ϕ is an abbreviation for $\rho \restriction \mathsf{fi}(\phi)$. Function composition is in applicative rather than diagrammatic order. Note that if ρ is a permissible environment for ϕ and X, then $f \circ \rho_\phi$ is a permissible environment for ϕ and Y. Considering cases, we have:

— Case tt:

$$X, \rho \models \mathrm{tt} \quad \text{iff} \quad Y, f \circ \rho_\phi \models \mathrm{tt}.$$

— Case $\neg\phi$:

$$\begin{aligned} X, \rho \models \neg\phi \quad &\text{iff} \quad X, \rho \not\models \phi \\ &\text{iff} \quad Y, f \circ \rho_\phi \not\models \phi \\ &\text{iff} \quad Y, f \circ \rho_{\neg\phi} \models \neg\phi. \end{aligned}$$

— Case $\phi_1 \wedge \phi_2$:

$$\begin{aligned} X, \rho \models \phi_1 \wedge \phi_2 \quad &\text{iff} \quad X, \rho \models \phi_1 \text{ and } X, \rho \models \phi_2 \\ &\text{iff} \quad Y, f \circ \rho_{\phi_1} \models \phi_1 \text{ and } Y, f \circ \rho_{\phi_2} \models \phi_2 \\ &\text{iff} \quad Y, f \circ \rho_{\phi_1 \wedge \phi_2} \models \phi_1 \text{ and } Y, f \circ \rho_{\phi_1 \wedge \phi_2} \models \phi_2 \text{ (using Lemma 4.11)} \\ &\text{iff} \quad Y, f \circ \rho_{\phi_1 \wedge \phi_2} \models \phi_1 \wedge \phi_2. \end{aligned}$$

https://doi.org/10.1017/S0960129513000510 Published online by Cambridge University Press

— Case $(x : a)\phi$:
Suppose $X, \rho \models (x : a)\phi$. Then there is $e \in X$ such that $\ell(e) = a$ and $X, \rho[x \mapsto e] \models \phi$. By the induction hypothesis,

$$Y, f \circ (\rho[x \mapsto e])_\phi \models \phi,$$

so

$$Y, (f \circ \rho_{(x:a)\phi})[x \mapsto f(e)] \models \phi$$

and

$$\ell(f(e)) = a.$$

Hence

$$Y, f \circ \rho_{(x:a)\phi} \models (x : a)\phi.$$

Conversely, if

$$Y, f \circ \rho_{(x:a)\phi} \models (x : a)\phi,$$

then

$$X, \rho \models (x : a)\phi.$$

— Case $\langle\langle x \rangle\phi$:
Suppose $X, \rho \models \langle\langle x \rangle\phi$. Then $X \overset{e}{\rightsquigarrow}_{\mathcal{C}} X'$ with $\rho(x) = e$ and $X', \rho \models \phi$. Let

$$\begin{aligned} e' &= f(e) \\ Y' &= Y \setminus \{e'\} \\ f' &= f \setminus \{(e, e')\}. \end{aligned}$$

Then $Y \overset{e'}{\rightsquigarrow}_{\mathcal{D}} Y'$ and $f' : X' \cong Y'$. By the induction hypothesis,

$$Y', f' \circ \rho_\phi \models \phi,$$

so

$$Y, f \circ \rho_{\langle\langle x \rangle\phi} \models \phi$$

and

$$Y, f \circ \rho_{\langle\langle x \rangle\phi} \models \langle\langle x \rangle\phi$$

as required.
Conversely, if

$$Y, f \circ \rho_{\langle\langle x \rangle\phi} \models \langle\langle x \rangle\phi,$$

then

$$X, \rho \models \langle\langle x \rangle\phi. \qquad \square$$

Note that Lemma 5.3 does not hold for any larger natural sublogic of EIL. This is because $\mathrm{EIL}_{\mathrm{ro}}$ contains all operators of EIL apart from $\langle x : a \rangle\rangle\phi$, and the induction fails for this case; two isomorphic configurations X and Y will not necessarily have the same possible forward transitions, since those potentially take us outside X and Y.

https://doi.org/10.1017/S0960129513000510 Published online by Cambridge University Press

Given any configuration X, we can create a closed formula $\theta_X \in \mathrm{EIL_{ro}}$ that gives the order structure of X. We make this precise in the following lemma.

Lemma 5.4. Let X be a configuration of a stable configuration structure $\mathcal{C}$. There is a *closed* formula $\theta_X \in \mathrm{EIL_{ro}}$ such that if Y is any configuration of a stable configuration structure $\mathcal{D}$ and $|Y| = |X|$, then $Y \cong X$ if and only if $Y \models \theta_X$.

Proof. Let $|X| = n$ and the events of X be enumerated as $\{e_1, \ldots, e_n\}$ in such a way that if $e_i <_X e_j$, then $i < j$: this is always possible. Let $\ell(e_i) = a_i$ $(i = 1, \ldots, n)$ and let $z_1, \ldots, z_n$ be distinct identifiers. Let ρ_X be the environment mapping z_i to e_i $(i = 1, \ldots, n)$.

The formula θ_X will use the identifiers z_i to express the ordering $<_X$. For each $k = 1, \ldots, n$, we shall define a formula θ_X^k that will, in effect, state whether $e_j <_X e_k$ for each $j \neq k$. The idea is as follows:

— If $e_j \not<_X e_k$, it is possible to reverse e_j without reversing e_k. Of course, we might have to reverse other events that are caused by e_j first. These events must have subscripts greater than j. So to reverse all events that do not cause e_k, we should start by reversing events with the highest subscript and work downwards.

— On the other hand, if $e_j <_X e_k$, it is impossible to reverse e_j without first reversing e_k, even if we have first reversed all events not causing e_k.

So we let θ_X^k be the formula obtained by:

(1) reversing $z_n, \ldots, z_{k+1}$;

(2) not reversing z_k;

(3) reversing as many as possible of $z_{k-1}, \ldots, z_1$, starting with z_{k-1} and working down to z_1 – call these $z_{i_1}, \ldots, z_{i_{r_k}}$; and finally

(4) stating that it is impossible to reverse the remaining members of $z_{k-1}, \ldots, z_1$ – these are precisely those j such that $e_j <_X e_k$, as discussed above.

Thus

$$\theta_X^k \stackrel{\mathrm{df}}{=} \langle\!\langle z_n \rangle \cdots \langle\!\langle z_{k+1} \rangle \langle\!\langle z_{i_1} \rangle \cdots \langle\!\langle z_{i_{r_k}} \rangle \bigwedge_{e_j <_X e_k} \neg \langle\!\langle z_j \rangle \mathrm{tt}$$

and

$$\theta_X^0 \stackrel{\mathrm{df}}{=} \langle\!\langle z_n \rangle \langle\!\langle z_{n-1} \rangle \cdots \langle\!\langle z_1 \rangle \mathrm{tt}.$$

Now let

$$\theta'_X \stackrel{\mathrm{df}}{=} \bigwedge_{k=0}^{n} \theta_X^k.$$

It is clear that $\theta'_X \in \mathrm{EIL_{dfro}}$ and $X, \rho_X \models \theta'_X$. Finally, let

$$\theta_X \stackrel{\mathrm{df}}{=} (z_1 : a_1) \cdots (z_n : a_n) \theta'_X.$$

Again it is clear that $\theta_X \in \mathrm{EIL_{ro}}$ and $X \models \theta_X$. Note that if $n = 0$, then $\theta_X = \mathrm{tt}$.

Now suppose $|Y| = |X|$ and $Y \models \theta_X$. Then there is ρ such that $Y, \rho \models \theta'_X$. We know that ρ assigns the n identifiers of θ'_X to different events of Y (since $Y, \rho \models \theta_X^0$), so ρ is onto Y. Let $e'_i = \rho(z_i)$ $(i = 1, \ldots, n)$. Then $\ell(e'_i) = a_i = \ell(e_i)$.

Take any $k \leqslant n$. We have $Y, \rho \models \theta_X^k$. We claim that for all j, we have $e'_j <_Y e'_k$ if and only if $e_j <_X e_k$. If $e_j \not<_X e_k$, then $Y, \rho \models \theta_X^k$ tells us that e'_j can be reversed without

https://doi.org/10.1017/S0960129513000510 Published online by Cambridge University Press

reversing e'_k. So $e'_j \not<_Y e'_k$. Conversely, suppose $e_j <_X e_k$. By $Y, \rho \models \theta^k_X$, we can reverse all $e'_{j'}$ such that $e_{j'} \not<_X e_k$ without reversing e'_k. This takes us to the configuration

$$Y_k \stackrel{\text{df}}{=} \{e'_i : e_i \leqslant_X e_k\}$$

with $e'_j \in Y_k$. Now $Y, \rho \models \theta^k_X$ further tells us that it is impossible to reverse any $e'_{j'}$ such that $e_{j'} <_X e_k$ without first reversing e'_k. So any sub-configuration of Y_k that contains e'_k must include the whole of Y_k, and thus, in particular, e'_j. This means that $e'_j <_{Y_k} e'_k$, which implies $e'_j <_Y e'_k$ as required. This completes the proof of the claim. It follows from the claim that $Y \cong X$ via the isomorphism $f(e_i) = e'_i$.

Conversely, suppose that $Y \cong X$ via the isomorphism $f : X \to Y$. Since $X \models \theta_X$, we now have $Y \models \theta_X$ by Lemma 5.3. □

Example 5.5. Consider an event structure with events e_1, e_2, e_3 and e_4 all labelled a. Assume that there is no conflict, $e_1 < e_3$ and $e_2 < \{e_3, e_4\}$, and that we are in the configuration X where all events have executed (from now on we will omit X from all formulas). The formulas θ^k implied by Lemma 5.4 are as follows:

$$\begin{aligned}
\theta^0 &\equiv \langle\!\langle z_4\rangle\!\langle\!\langle z_3\rangle\!\langle\!\langle z_2\rangle\!\langle\!\langle z_1\rangle \mathtt{tt} \\
\theta^1 &\equiv \langle\!\langle z_4\rangle\!\langle\!\langle z_3\rangle\!\langle\!\langle z_2\rangle \mathtt{tt} \\
\theta^2 &\equiv \langle\!\langle z_4\rangle\!\langle\!\langle z_3\rangle\!\langle\!\langle z_1\rangle \mathtt{tt} \\
\theta^3 &\equiv \langle\!\langle z_4\rangle(\neg\langle\!\langle z_2\rangle \mathtt{tt} \wedge \neg\langle\!\langle z_1\rangle \mathtt{tt}) \\
\theta^4 &\equiv \langle\!\langle z_3\rangle\!\langle\!\langle z_1\rangle(\neg\langle\!\langle z_2\rangle \mathtt{tt}).
\end{aligned}$$

Then $\theta' \equiv \bigwedge_{k=0}^4 \theta^k$ and the formula θ that characterises precisely the causal structure of X is

$$(z_1 : a)(z_2 : a)(z_3 : a)(z_4 : a)\theta'.$$

Remark 5.6. We can remove the condition $|Y| = |X|$ in Lemma 5.4 if we have a formula ζ that holds precisely in empty configurations. We can then amend θ_X by redefining θ^0_X to be

$$\langle\!\langle z_n\rangle\!\langle\!\langle z_{n-1}\rangle \cdots \langle\!\langle z_1\rangle \zeta.$$

If the set Act of labels is finite, we can set

$$\zeta \stackrel{\text{df}}{=} \bigwedge_{a \in \mathsf{Act}} \neg(x : a)\mathtt{tt}.$$

The next lemma follows fairly immediately from the proof of Lemma 5.4 together with Lemma 5.3.

Lemma 5.7. Let X be a configuration of a stable configuration structure $\mathcal{C}$. Let $\{z_e : e \in X\}$ be distinct identifiers. Let the environment ρ_X be defined by $\rho_X(z_e) = e$ $(e \in X)$. There is a formula $\theta'_X \in \mathrm{EIL}_{\mathrm{dfro}}$ with $\mathsf{fi}(\theta') = \{z_e : e \in X\}$ such that $X, \rho_X \models \theta'_X$ and if Y is any configuration of a stable configuration structure $\mathcal{D}$ and $|Y| = |X|$, then $Y \cong X$ if and only if $\exists \rho.\ Y, \rho \models \theta'_X$.

Proof. The proof is really already contained in the proof of Lemma 5.4.

https://doi.org/10.1017/S0960129513000510 Published online by Cambridge University Press

Let $|X| = n$, and let θ'_X, ρ_X be defined as in the proof of Lemma 5.4, except that we change z_i to z_{e_i} $(i = 1, \ldots, n)$. Then $X, \rho_X \models \theta'_X$. Also, if we take any Y with $|Y| = |X|$, and suppose $Y, \rho \models \theta'_X$, then we can deduce that $Y \cong X$.

Conversely, suppose $Y \cong X$ via the isomorphism $f : X \to Y$. Since $X, \rho_X \models \theta'_X$, we have $Y, \rho \models \theta'_X$ for some ρ by Lemma 5.3. □

5.2. *Logics for history-preserving bisimulations*

We start by showing that EIL characterises HH-bisimulation (Theorem 5.9). We then present sublogics of EIL corresponding to H-bisimulation, WH-bisimulation and HWH-bisimulation.

However, we shall begin with the following lemma before giving Theorem 5.9.

Lemma 5.8. Let X be a configuration of a stable configuration structure $\mathcal{C}$, and let $\phi \in$ EIL. Suppose σ maps $\mathsf{fi}(\phi)$ (not necessarily injectively) to a set of fresh identifiers (in particular, ones not occurring either free or bound in ϕ), ρ is an environment for ϕ and X, ρ' is an environment for $\sigma(\phi)$ and X, and for any $x \in \mathsf{fi}(\phi)$, we have $\rho(x) = \rho'(\sigma(x))$. Here $\sigma(\phi)$ is obtained by replacing each occurrence of a free identifier x in ϕ by $\sigma(x)$.

Then $X, \rho \models \phi$ if and only if $X, \rho' \models \sigma(\phi)$.

Proof. Note that we allow σ, ρ, ρ' to be non-injective, and that we effectively define $\sigma(\phi)$ by induction on ϕ during the course of the proof.

By induction on ϕ, we have:

— Case $\mathtt{tt}$:

$$X, \rho \models \mathtt{tt} \;\text{ iff }\; X, \rho' \models \sigma(\mathtt{tt}) = \mathtt{tt}.$$

— Case $\neg\phi$:

$$\begin{aligned} X, \rho \models \neg\phi \;&\text{iff}\; X, \rho \not\models \phi \\ &\text{iff}\; X, \rho' \not\models \sigma(\phi) \\ &\text{iff}\; X, \rho' \models \neg\sigma(\phi) = \sigma(\neg\phi). \end{aligned}$$

— Case $\phi_1 \wedge \phi_2$:

$$\begin{aligned} X, \rho \models \phi_1 \wedge \phi_2 \;&\text{iff}\; X, \rho \models \phi_1 \text{ and } X, \rho \models \phi_2 \\ &\text{iff}\; X, \rho' \models \sigma_1(\phi_1) \text{ and } X, \rho' \models \sigma_2(\phi_2) \\ &\text{iff}\; X, \rho' \models \sigma_1(\phi_1) \wedge \sigma_2(\phi_2) = \sigma(\phi_1 \wedge \phi_2) \end{aligned}$$

where $\sigma_i = \sigma \upharpoonright \mathsf{fi}(\phi_i)$ $(i = 1, 2)$.

— Case $\langle x : a \rangle\rangle\phi$:

$$\begin{aligned} X, \rho \models \langle x : a \rangle\rangle\phi \;&\text{iff}\; \exists X', e.\ X \xrightarrow{e} X', \ell(e) = a, X', \rho[x \mapsto e] \models \phi \\ &\text{iff}\; \exists X', e.\ X \xrightarrow{e} X', \ell(e) = a, X', \rho'[x \mapsto e] \models \sigma'(\phi) \\ &\text{iff}\; X, \rho' \models \langle x : a \rangle\rangle\sigma'(\phi) = \sigma(\langle x : a \rangle\rangle\phi) \end{aligned}$$

where $\sigma' = \sigma$ if $x \notin \mathsf{fi}(\phi)$, and $\sigma' = \sigma[x \mapsto x]$ if $x \in \mathsf{fi}(\phi)$.

https://doi.org/10.1017/S0960129513000510 Published online by Cambridge University Press

— Case $(x:a)\phi$:

$$\begin{aligned} X,\rho \models (x:a)\phi \ &\text{iff}\ \ \exists e \in X.\, \ell(e) = a \text{ and } X, \rho[x \mapsto e] \models \phi \\ &\text{iff}\ \ \exists e \in X.\, \ell(e) = a \text{ and } X, \rho'[x \mapsto e] \models \sigma'(\phi) \\ &\text{iff}\ \ X,\rho' \models (x:a)\sigma'(\phi) = \sigma((x:a)\phi) \end{aligned}$$

where, again, $\sigma' = \sigma$ if $x \notin \mathsf{fi}(\phi)$, and $\sigma' = \sigma[x \mapsto x]$ if $x \in \mathsf{fi}(\phi)$.

— Case $\langle\!\langle x \rangle\!\rangle\phi$:

$$\begin{aligned} X,\rho \models \langle\!\langle x \rangle\!\rangle\phi \ &\text{iff}\ \ \exists X', e.\ X \overset{e}{\rightsquigarrow} X', \rho(x) = e \text{ and } X', \rho \models \phi \\ &\text{iff}\ \ \exists X', e.\ X \overset{e}{\rightsquigarrow} X', \rho'(\sigma(x)) = e \text{ and } X', \rho' \models \sigma'(\phi) \\ &\text{iff}\ \ X,\rho' \models \langle\!\langle \sigma(x) \rangle\!\rangle\sigma'(\phi) = \sigma(\langle\!\langle x \rangle\!\rangle\phi) \end{aligned}$$

where $\sigma' = \sigma \setminus x$ if $x \notin \mathsf{fi}(\phi)$, and $\sigma' = \sigma$ if $x \in \mathsf{fi}(\phi)$. □

The next result is related to the result of Nielsen and Clausen (1994b) stating that a logic with reverse event index modality (discussed in Section 2 above) characterises HH.

Theorem 5.9. Let $\mathcal{C}$ and $\mathcal{D}$ be stable configuration structures. Then, $\mathcal{C} \approx_{\mathsf{hh}} \mathcal{D}$ if and only if $\mathcal{C} \sim_{\mathrm{EIL}} \mathcal{D}$.

Proof.

($\Rightarrow$) Let $\mathcal{R}$ be an HH bisimulation between $\mathcal{C}$ and $\mathcal{D}$. We shall show by induction on ϕ that for all X, Y, f, if $\mathcal{R}(X, Y, f)$, then for all $\phi \in$ EIL and all ρ (a permissible environment for ϕ and X), we have $X, \rho \models \phi$ if and only if $Y, f \circ \rho_\phi \models \phi$. Recall that ρ_ϕ is an abbreviation for $\rho \restriction \mathsf{fi}(\phi)$ and that if ρ is a permissible environment for ϕ and X, then $f \circ \rho_\phi$ is a permissible environment for ϕ and Y.
By considering initial (empty) configurations, our induction hypothesis implies that $\mathcal{C} \sim_{\mathrm{EIL}} \mathcal{D}$.
So, supposing $\mathcal{R}(X, Y, f)$, we have:

— Case tt:
It is clear that

$$X, \rho \models \mathsf{tt} \ \text{ iff } \ Y, f \circ \rho_{\mathsf{tt}} \models \mathsf{tt}.$$

— Case $\neg\phi$:

$$\begin{aligned} X,\rho \models \neg\phi \ &\text{iff}\ \ X,\rho \not\models \phi \\ &\text{iff}\ \ Y, f \circ \rho_\phi \not\models \phi \qquad \text{(using the induction hypothesis)} \\ &\text{iff}\ \ Y, f \circ \rho_{\neg\phi} \not\models \phi \\ &\text{iff}\ \ Y, f \circ \rho_{\neg\phi} \models \neg\phi. \end{aligned}$$

https://doi.org/10.1017/S0960129513000510 Published online by Cambridge University Press

— Case $\phi_1 \wedge \phi_2$:

$$\begin{aligned} X, \rho \models \phi_1 \wedge \phi_2 \;\; &\text{iff} \;\; X, \rho \models \phi_1 \text{ and } X, \rho \models \phi_2 \\ &\text{iff} \;\; Y, f \circ \rho_{\phi_1} \models \phi_1 \text{ and } Y, f \circ \rho_{\phi_2} \models \phi_2 \\ &\qquad\qquad\qquad\qquad\qquad \text{(using the induction hypothesis)} \\ &\text{iff} \;\; Y, f \circ \rho_{\phi_1 \wedge \phi_2} \models \phi_1 \text{ and } Y, f \circ \rho_{\phi_1 \wedge \phi_2} \models \phi_2 \\ &\qquad\qquad\qquad\qquad\qquad \text{(using Lemma 4.11)} \\ &\text{iff} \;\; Y, f \circ \rho_{\phi_1 \wedge \phi_2} \models \phi_1 \wedge \phi_2. \end{aligned}$$

— Case $\langle x : a \rangle\!\rangle \phi$:
Suppose $X, \rho \models \langle x : a \rangle\!\rangle \phi$.
Then $X \xrightarrow{e}_{\mathcal{C}} X'$ for some X', e such that $\ell(e) = a$ and $X', \rho[x \mapsto e] \models \phi$.
Since $\mathcal{R}(X, Y, f)$, there are Y', e', f' such that

$$\begin{gathered} \ell(e') = a \\ Y \xrightarrow{e'}_{\mathcal{D}} Y' \\ \mathcal{R}(X', Y', f') \\ f' = f \cup \{(e, e')\}. \end{gathered}$$

By the induction hypothesis,

$$Y', f' \circ (\rho[x \mapsto e])_\phi \models \phi.$$

Hence

$$Y', (f \circ \rho_{\langle x:a \rangle\!\rangle \phi})[x \mapsto e'] \models \phi.$$

So

$$Y, f \circ \rho_{\langle x:a \rangle\!\rangle \phi} \models \langle x : a \rangle\!\rangle \phi$$

as required.
The converse is similar.

— Case $(x : a)$:
Suppose $X, \rho \models (x : a)\phi$.
Then there is $e \in X$ such that $\ell(e) = a$ and $X, \rho[x \mapsto e] \models \phi$. By the induction hypothesis,

$$Y, f \circ (\rho[x \mapsto e])_\phi \models \phi.$$

So

$$Y, (f \circ \rho_{(x:a)\phi})[x \mapsto f(e)] \models \phi.$$

It is clear that $\ell(f(e)) = a$, so

$$Y, f \circ \rho_{(x:a)\phi} \models (x : a)\phi.$$

The converse is similar.

— Case $\langle\!\langle x \rangle \phi$:
Suppose $X, \rho \models \langle\!\langle x \rangle \phi$ and let $e = \rho(x)$ and $X' = X \setminus \{e\}$.

https://doi.org/10.1017/S0960129513000510 Published online by Cambridge University Press

Then $X \overset{e}{\rightsquigarrow}_{\mathcal{C}} X'$ and $X', \rho \models \phi$.
Since $\mathcal{R}(X, Y, f)$, we get Y', e', f' such that

$$Y \overset{e'}{\rightsquigarrow}_{\mathcal{D}} Y'$$
$$\mathcal{R}(X', Y', f')$$
$$f' = f \setminus \{(e, e')\}.$$

By the induction hypothesis,

$$Y', f' \circ \rho_{\phi} \models \phi.$$

So

$$Y', f \circ \rho_{\langle\langle x \rangle\rangle \phi} \models \phi.$$

Hence

$$Y, f \circ \rho_{\langle\langle x \rangle\rangle \phi} \models \langle\langle x \rangle\rangle \phi$$

as required.
The converse is similar.

($\Leftarrow$) Suppose $\mathcal{C} \sim_{\mathrm{EIL}} \mathcal{D}$. Define $\mathcal{R}(X, Y, f)$ if and only if:

— f is an order isomorphism between X and Y;

— for any $\phi \in$ EIL and ρ (a permissible environment for ϕ and X) with $\mathsf{rge}(\rho) \subseteq X$, we have $X, \rho \models \phi$ if and only if $Y, f \circ \rho \models \phi$.
(Note that by considering negated formulas, $X, \rho \models \phi$ if and only if $Y, f \circ \rho \models \phi$ is equivalent to $X, \rho \models \phi$ implies $Y, f \circ \rho \models \phi$.)

We shall now show that $\mathcal{R}$ is an HH bisimulation. It is clear that $\mathcal{R}(\emptyset, \emptyset, \emptyset)$ since $\mathcal{C} \sim_{\mathrm{EIL}} \mathcal{D}$, so assuming $\mathcal{R}(X, Y, f)$, we have:

(1) Suppose $X \xrightarrow{e}_{\mathcal{C}} X'$ with $\ell(e) = a$, and for all e', Y' such that $Y \xrightarrow{e'}_{\mathcal{D}} Y'$ with $\ell(e') = a$, we have $\neg\mathcal{R}(X', Y', f')$, where $f' = f \cup \{(e, e')\}$. There are only finitely many such e' due to the image-finiteness of our configuration structures. Let all such e', Y', f' be e_i, Y_i, f_i for $i \in I$. For each i, since $\neg\mathcal{R}(X', Y_i, f_i)$, at least one of the following holds:

(a) there are ϕ_i, ρ_i with $\mathsf{rge}(\rho_i) \subseteq X'$ such that $X', \rho_i \models \phi_i$ and $Y_i, f_i \circ \rho_i \not\models \phi_i$;

(b) f_i is not an order isomorphism between X' and Y_i.

Let $\{z_{e'} : e' \in X'\}$ be a set of fresh distinct identifiers. Let the environment $\rho_{X'}$ be defined by $\rho_{X'}(z_{e'}) = e'$ (all $e' \in X'$). We are going to standardise all formulas to use this environment so that we can conjoin them. Similarly, let $\rho_X = \rho_{X'} \setminus z_e$. In each of the cases (a) and (b), we shall obtain ψ_i such that $X', \rho_{X'} \models \psi_i$ and $Y_i, f_i \circ \rho_{X'} \not\models \psi_i$:

(a) We have $X', \rho_i \models \phi_i$ and $Y_i, f_i \circ \rho_i \not\models \phi_i$. Let σ_i be defined by $\sigma_i(x) = z_{\rho_i(x)}$ for $x \in \mathsf{fi}(\phi_i)$. Let $\psi_i = \sigma_i(\phi_i)$, which is obtained by replacing each free identifier x in ϕ_i by $\sigma_i(x)$. It is clear that

$$\rho_i(x) = \rho_{X'}(\sigma_i(x))$$

for each $x \in \mathsf{fi}(\phi_i)$. Then $X', \rho_{X'} \models \psi_i$ by Lemma 5.8. Similarly,

$$f_i \circ \rho_i(x) = f_i \circ \rho_{X'}(\sigma_i(x))$$

for each $x \in \mathsf{fi}(\phi_i)$, so

$$Y_i, f_i \circ \rho_{X'} \not\models \psi_i,$$

again by Lemma 5.8.

(b) Let $\psi_i \stackrel{\mathrm{df}}{=} \theta'_{X'}$ as in Lemma 5.7. Then $X', \rho_{X'} \models \psi_i$, by Lemma 5.7. Also, $Y_i, f_i \circ \rho_{X'} \not\models \psi_i$, again by Lemma 5.7, noting that $|Y_i| = |X'|$.
Let Ψ be $\bigwedge_{i \in I} \psi_i$. It is clear that $X', \rho_{X'} \models \Psi$, that is,

$$X', \rho_X[z_e \mapsto e] \models \Psi,$$

so

$$X, \rho_X \models \langle z_e : a \rangle\!\rangle \Psi.$$

Also, for each $i \in I$, we have

$$Y_i, f_i \circ \rho_{X'} \not\models \Psi,$$

that is,

$$Y_i, (f \circ \rho_X)[z_e \mapsto e_i] \not\models \Psi.$$

Hence,

$$Y, f \circ \rho_X \not\models \langle z_e : a \rangle\!\rangle \Psi,$$

which contradicts $\mathcal{R}(X, Y, f)$.

(2) The case where $Y \xrightarrow{e}_{\mathcal{D}} Y'$ is similar to the previous case.

(3) Suppose $X \overset{e}{\rightsquigarrow}_{\mathcal{C}} X'$. We must show that

$$Y \overset{f(e)}{\rightsquigarrow}_{\mathcal{D}} Y' = Y \setminus \{f(e)\}$$

and $\mathcal{R}(X', Y', f')$, where $f' = f \restriction X'$. It is clear that f' is an order isomorphism between X' and Y'. To establish $Y \overset{f(e)}{\rightsquigarrow}_{\mathcal{D}} Y'$, note that

$$X, [z \mapsto e] \models \langle\!\langle z \rangle \mathsf{tt}.$$

Hence

$$Y, f \circ [z \mapsto e] \models \langle\!\langle z \rangle \mathsf{tt}.$$

Suppose there are ϕ, ρ such that $X', \rho \models \phi$ but $Y', f' \circ \rho \not\models \phi$. Let z be fresh. Then

$$X, \rho[z \mapsto e] \models \langle\!\langle z \rangle \phi$$

but

$$Y, f \circ (\rho[z \mapsto e]) \not\models \langle\!\langle z \rangle \phi,$$

since

$$f \circ (\rho[z \mapsto e]) = (f' \circ \rho)[z \mapsto f(e)]),$$

which contradicts $\mathcal{R}(X, Y, f)$.

https://doi.org/10.1017/S0960129513000510 Published online by Cambridge University Press

(4) The case where $Y \overset{e}{\rightsquigarrow}_{\mathcal{D}} Y'$ is similar to the previous case. □

Remark 5.10. The proof of Theorem 5.9 would still work with the logic restricted by not using declarations $(x : a)\phi$, since they are not used in the ($\Leftarrow$) direction. However, we include declarations in EIL because they are useful in defining sublogics for WH, among other things.

We now define a sublogic of EIL that characterises history-preserving bisimulation.

Definition 5.11. $\mathrm{EIL_h}$ is given as follows, where ϕ_r is a formula of $\mathrm{EIL_{ro}}$:

$$\phi ::= \mathtt{tt} \mid \neg\phi \mid \phi \wedge \phi' \mid \langle x : a\rangle\rangle\phi \mid (x : a)\phi \mid \phi_r.$$

$\mathrm{EIL_h}$ is just EIL with $\langle\langle x : a\rangle\phi$ replaced by $\phi_r \in \mathrm{EIL_{ro}}$. Thus we are not allowed to go forward after going in reverse. This concept of disallowing forward moves embedded inside reverse moves appears in Goltz *et al.* (1992).

Theorem 5.12. Let $\mathcal{C}$ and $\mathcal{D}$ be stable configuration structures. Then, $\mathcal{C} \approx_{\mathrm{h}} \mathcal{D}$ if and only if $\mathcal{C} \sim_{\mathrm{EIL_h}} \mathcal{D}$.

Proof. We adapt the proof of Theorem 5.9 as follows:

($\Rightarrow$) Let $\mathcal{R}$ be an H bisimulation between $\mathcal{C}$ and $\mathcal{D}$. We show by induction on ϕ that for all X, Y, f, if $\mathcal{R}(X, Y, f)$, then for all $\phi \in \mathrm{EIL_h}$ and all ρ (environment for ϕ and X), we have $X, \rho \models \phi$ if and only if $Y, f \circ \rho \models \phi$.
The cases for tt, negation, conjunction, $\langle x : a\rangle\rangle\phi$ and $(x : a)\phi$ are as in the proof of Theorem 5.9. This only leaves the case of $\phi_r \in \mathrm{EIL_{ro}}$. For this case, instead of using the main induction hypothesis, we use Lemma 5.3.

($\Leftarrow$) Suppose $\mathcal{C} \sim_{\mathrm{EIL_h}} \mathcal{D}$. Define $\mathcal{R}(X, Y, f)$ if and only if $f : X \cong Y$ and for any $\phi \in \mathrm{EIL_h}$ and any ρ (an environment for ϕ and X) we have $X, \rho \models \phi$ if and only if $Y, f \circ \rho \models \phi$. We shall show that $\mathcal{R}$ is an H bisimulation.
The proof is the same as the part for forward transitions in the proof of Theorem 5.9. We just need to note that each ψ_i as well as Ψ and $\langle z_e : a\rangle\rangle\Psi$ are formulas of $\mathrm{EIL_h}$. □

Remark 5.13. Just as for Theorem 5.9, Theorem 5.12 would still hold if we disallowed declarations $(x : a)\phi$. This gives the following more minimal logic, where $\phi_r \in \mathrm{EIL_{dfro}}$:

$$\phi ::= \mathtt{tt} \mid \neg\phi \mid \phi \wedge \phi' \mid \langle x : a\rangle\rangle\phi \mid \phi_r.$$

We next define a sublogic $\mathrm{EIL_{wh}}$ of $\mathrm{EIL_h}$ that characterises weak history-preserving bisimulation. We get from $\mathrm{EIL_h}$ to $\mathrm{EIL_{wh}}$ by simply requiring that all formulas of $\mathrm{EIL_{wh}}$ are *closed*.

Definition 5.14. $\mathrm{EIL_{wh}}$ is given as follows, where ϕ_{rc} is a *closed* formula of $\mathrm{EIL_{ro}}$ (Definition 5.2):

$$\phi ::= \mathtt{tt} \mid \neg\phi \mid \phi \wedge \phi' \mid \langle a\rangle\rangle\phi \mid \phi_{rc}.$$

https://doi.org/10.1017/S0960129513000510 Published online by Cambridge University Press

In the above definition, we write $\langle a\rangle\rangle\phi$ rather than $\langle x : a\rangle\rangle\phi$ since ϕ is closed and, in particular, x does not occur free in ϕ (Notation 4.6). Also, we omit declarations $(x : a)\phi$ since they have no effect when ϕ is closed. Of course, declarations can occur in ϕ_{rc}.

Theorem 5.15. Let $\mathcal{C}$ and $\mathcal{D}$ be stable configuration structures. Then $\mathcal{C} \approx_{\mathsf{wh}} \mathcal{D}$ if and only if $\mathcal{C} \sim_{\mathrm{EIL_{wh}}} \mathcal{D}$.

Proof. We can take all environments to be empty since we are dealing with closed formulas. Apart from the use of Lemmas 5.3 and 5.4 to handle formulas of $\mathrm{EIL_{ro}}$, the proof is much as for standard Hennessy–Milner logic (Hennessy and Milner 1985):

($\Rightarrow$) Let $\mathcal{R}$ be a WH bisimulation between $\mathcal{C}$ and $\mathcal{D}$. We show by induction on ϕ that for all X, Y, if $\mathcal{R}(X, Y)$, then $X \models \phi$ if and only if $Y \models \phi$.
So we suppose $\mathcal{R}(X, Y)$. Then, considering cases:

— Cases tt, negation and conjunction:
These are all straightforward.

— Case $X \models \langle a\rangle\rangle\phi$:
Then for some X', we have $X \xrightarrow{a} X'$, and $X' \models \phi$. So there is some Y' such that $Y \xrightarrow{a} Y'$ and $\mathcal{R}(X', Y')$. By the induction hypothesis, $Y' \models \phi$. Hence $Y \models \langle a\rangle\rangle\phi$ as required.
The converse where $Y \models \langle a\rangle\rangle\phi$ is similar.

— Case $\phi_{rc} \in \mathrm{EIL_{ro}}$:
This follows from Lemma 5.3, noting that $X \cong Y$.

($\Leftarrow$) Suppose $\mathcal{C} \sim_{\mathrm{EIL_{wh}}} \mathcal{D}$. Define $\mathcal{R}(X, Y)$ if and only if both $X \cong Y$ and for any $\phi \in \mathrm{EIL_{wh}}$ we have $X \models \phi$ if and only if $Y \models \phi$. We shall show that $\mathcal{R}$ is a WH bisimulation.
We proceed in a similar manner to the ($\Leftarrow$) direction in the proof of Theorem 5.9, though the details are different.
It is clear that $\mathcal{R}(\emptyset, \emptyset)$, since $\mathcal{C} \sim_{\mathrm{EIL_{wh}}} \mathcal{D}$, so we assume $\mathcal{R}(X, Y)$.
Suppose $X \xrightarrow{a}_{\mathcal{C}} X'$, and that for all Y' such that $Y \xrightarrow{a}_{\mathcal{D}} Y'$, we have $\neg\mathcal{R}(X', Y')$. There are only finitely many such Y'. Let all such Y' be Y_i for $i \in I$. For each i, since $\neg\mathcal{R}(X', Y_i)$, one of the following holds:

(1) $X' \ncong Y_i$.

(2) There is ψ_i such that $X' \models \psi_i$ and $Y_i \not\models \psi_i$.

In case (1), let ψ_i be $\theta_{X'}$ as in Lemma 5.4. It is clear that $X' \models \psi_i$. Since $|Y_i| = |X'|$ (both obtained by adding one event to isomorphic configurations Y, X), it must be that $Y_i \not\models \psi_i$.
Thus, for each of cases (1) and (2) we have a formula ψ_i of $\mathrm{EIL_{wh}}$ such that $X' \models \psi_i$ and $Y_i \not\models \psi_i$.
Let Ψ be $\bigwedge_{i\in I} \psi_i$. It is clear that $X' \models \Psi$, so $X \models \langle a\rangle\rangle\Psi$. Also, for each $i \in I$, we have $Y_i \not\models \Psi$. Hence, $Y \not\models \langle a\rangle\rangle\Psi$, which contradicts $\mathcal{R}(X, Y)$.
The case where $Y \xrightarrow{a}_{\mathcal{D}} Y'$ is similar to that for $X \xrightarrow{a}_{\mathcal{C}} X'$. □

https://doi.org/10.1017/S0960129513000510 Published online by Cambridge University Press

We believe that $\mathrm{EIL_{wh}}$ is the first logic proposed for weak history-preserving bisimulation with autoconcurrency allowed. Goltz *et al.* (1992) described a logic for weak history-preserving bisimulation with no autoconcurrency allowed, but in that case, weak history-preserving bisimulation is as strong as history-preserving bisimulation (van Glabbeek and Goltz 2001).

Just as we weakened $\mathrm{EIL_h}$ to get $\mathrm{EIL_{wh}}$, we can weaken EIL by requiring that forward transitions $\langle x : a \rangle\rangle \phi$ are only allowed if ϕ is closed. Again, instead of $\langle x : a \rangle\rangle \phi$, we write $\langle a \rangle\rangle \phi$. This gives us $\mathrm{EIL_{hwh}}$.

Definition 5.16. $\mathrm{EIL_{hwh}}$ is given below, where ϕ_c ranges over closed formulas of $\mathrm{EIL_{hwh}}$:

$$\phi ::= \mathrm{tt} \mid \neg\phi \mid \phi \wedge \phi' \mid \langle a \rangle\rangle \phi_c \mid (x : a)\phi \mid \langle\langle x \rangle \phi.$$

$\mathrm{EIL_{wh}}$ is clearly a sublogic of $\mathrm{EIL_{hwh}}$ as well as of $\mathrm{EIL_h}$.

Theorem 5.17. Let $\mathcal{C}$ and $\mathcal{D}$ be stable configuration structures. Then, $\mathcal{C} \approx_{\mathsf{hwh}} \mathcal{D}$ if and only if $\mathcal{C} \sim_{\mathrm{EIL_{hwh}}} \mathcal{D}$.

Proof.

($\Rightarrow$) Let $\mathcal{R}$ be an HWH bisimulation between $\mathcal{C}$ and $\mathcal{D}$. We show by induction on ϕ that for all X, Y, f, if $\mathcal{R}(X, Y, f)$, then for all $\phi \in \mathrm{EIL_{hwh}}$ and all ρ (a permissible environment for ϕ and X), we have $X, \rho \models \phi$ if and only if $Y, f \circ \rho_\phi \models \phi$. Recall that ρ_ϕ is an abbreviation for $\rho \restriction \mathsf{fi}(\phi)$.

All cases apart from $\langle a \rangle\rangle \phi_c$ are the same as in the proof of Theorem 5.9, and the $\langle a \rangle\rangle \phi_c$ case is the same as in the proof of Theorem 5.15.

($\Leftarrow$) Suppose $\mathcal{C} \sim_{\mathrm{EIL_{hwh}}} \mathcal{D}$ and define $\mathcal{R}(X, Y, f)$ if and only if:

— f is an order isomorphism between X and Y;

— for any $\phi \in \mathrm{EIL_{hwh}}$ and ρ (a permissible environment for both ϕ and X) with $\mathsf{rge}(\rho) \subseteq X$, we have $X, \rho \models \phi$ if and only if $Y, f \circ \rho \models \phi$.
(Note that by considering negated formulas, $X, \rho \models \phi$ if and only if $Y, f \circ \rho \models \phi$ is equivalent to $X, \rho \models \phi$ implies $Y, f \circ \rho \models \phi$.)

We show that $\mathcal{R}$ is an HWH bisimulation. It is clear that $\mathcal{R}(\emptyset, \emptyset, \emptyset)$ since we have $\mathcal{C} \sim_{\mathrm{EIL_{hwh}}} \mathcal{D}$, so we assume $\mathcal{R}(X, Y, f)$:

(1) Suppose $X \xrightarrow{e}_{\mathcal{C}} X'$ with $\ell(e) = a$. We must show that there are e', Y', f' such that $Y \xrightarrow{e'}_{\mathcal{D}} Y'$ with $\ell(e') = a$, and $\mathcal{R}(X', Y', f')$.

We now suppose, in order to show a contradiction, that there are no such e', Y', f'.

Let all e', Y', f' such that $Y \xrightarrow{e'}_{\mathcal{D}} Y'$, $\ell(e') = a$ and $f' : X' \cong Y'$ be enumerated as e_i, Y_i, f_i for $i \in I$. There are only finitely many such e', Y', f'. Note that for a given e' there is only one $Y' = Y \cup \{e_i\}$, but there may be more than one possible isomorphism $f' : X' \cong Y'$.

For each $i \in I$, since $\neg\mathcal{R}(X_i, Y_i, f_i)$, there are ϕ_i, ρ_i with $\mathsf{rge}(\rho_i) \subseteq X'$ such that $X', \rho_i \models \phi_i$ and $Y_i, f_i \circ \rho_i \not\models \phi_i$.

https://doi.org/10.1017/S0960129513000510 Published online by Cambridge University Press

Let $\{z_{e'} : e' \in X'\}$ be a set of fresh distinct identifiers. Let the environment $\rho_{X'}$ be defined by $\rho_{X'}(z_{e'}) = e'$ (all $e' \in X'$). We are going to standardise all formulas to use this environment so that we can conjoin them. Similarly, let $\rho_X = \rho_{X'} \setminus z_e$.

We shall obtain ψ_i such that $X', \rho_{X'} \models \psi_i$ and $Y_i, f_i \circ \rho_{X'} \not\models \psi_i$.

Let σ_i be defined by $\sigma_i(x) = z_{\rho_i(x)}$ for $x \in \mathsf{fi}(\phi_i)$. Let $\psi_i = \sigma_i(\phi_i)$, which is obtained by replacing each free identifier x in ϕ_i by $\sigma_i(x)$. It is clear that

$$\rho_i(x) = \rho_{X'}(\sigma_i(x))$$

for each $x \in \mathsf{fi}(\phi_i)$. Then $X', \rho_{X'} \models \psi_i$ by Lemma 5.8. Similarly,

$$f_i \circ \rho_i(x) = f_i \circ \rho_{X'}(\sigma_i(x))$$

for each $x \in \mathsf{fi}(\phi_i)$, so $Y_i, f_i \circ \rho_{X'} \not\models \psi_i$, again by Lemma 5.8.

Let $\theta'_{X'}$ be as in Lemma 5.7. The environment $\rho_{X'}$ we use here is taken to be the same as the one in the statement of Lemma 5.7. Thus $X', \rho_{X'} \models \theta'_{X'}$ by Lemma 5.7.

Let $\Psi' \stackrel{\text{df}}{=} \theta'_{X'} \wedge \bigwedge_{i \in I} \psi_i$. It is clear that $X', \rho_{X'} \models \Psi'$.

We now close Φ' by declaring all identifiers $z_{e'}$ $(e' \in X')$. Let

$$\Psi \stackrel{\text{df}}{=} (z_{e'} : \ell(e'))_{e' \in X'} \Psi',$$

using an obvious notation. We now have $X' \models \Psi$, and thus $X \models \langle a \rangle\rangle \Psi$, with $\langle a \rangle\rangle \Psi \in \mathrm{EIL}_{\mathrm{hwh}}$.

Since $\mathcal{R}(X, Y, f)$, we must have $Y \models \langle a \rangle\rangle \Psi$. So there are e' and Y' such that $Y \xrightarrow{e'}_{\mathcal{D}} Y'$, $\ell(e') = a$ and $Y' \models \Psi$. There is an environment ρ' with

$$\mathsf{dom}(\rho') = \{z_e : e \in X'\},$$

such that $Y', \rho' \models \Psi'$. In particular, $Y', \rho' \models \theta'_{X'}$. Since $|Y'| = |X'|$, we have $f' : X' \cong Y'$ where $f(e) = \rho'(z_e)$ (by Lemma 5.7 and the proof of Lemma 5.4).

But then e', Y', f' must be e_i, Y_i, f_i for some $i \in I$. So

$$Y', f' \circ \rho_{X'} \not\models \psi_i.$$

But for each $e \in X'$, we have

$$f' \circ \rho_{X'}(x_e) = f'(e) = \rho'(z_e).$$

So $Y', \rho' \not\models \psi_i$, which contradicts $Y', \rho' \models \Psi'$.

(2) The case where $Y \xrightarrow{e}_{\mathcal{D}} Y'$ is similar to the previous case.

(3) The case where $X \stackrel{e}{\rightsquigarrow}_{\mathcal{C}} X'$ is similar to the corresponding case in the proof of Theorem 5.9.

(4) The case where $Y \stackrel{e}{\rightsquigarrow}_{\mathcal{D}} Y'$ is similar to the previous case. □

With no (equidepth) autoconcurrency, we know that $\approx_{\mathsf{hwh}}$ is as strong as $\approx_{\mathsf{hh}}$ (Bednarczyk 1991; Phillips and Ulidowski 2012), so $\mathrm{EIL}_{\mathrm{hwh}}$ is as strong as EIL in this case.

https://doi.org/10.1017/S0960129513000510 Published online by Cambridge University Press

5.3. *Logics for pomset and step bisimulation*

We conclude our investigation of the sublogics of EIL characterising various equivalences by looking at the four remaining equivalences from Figure 1, namely PB, WHPB, SB and IB. Logics for PB and SB have already been presented by Baldan and Crafa, and our logics for these have similarities with theirs.

Baldan and Crafa's logic for PB uses the idea that we are not allowed to apply $\neg$ or $\wedge$ to open formulas. This means that we cannot branch using $\wedge$ into two different futures and use causal information from the past. We can adapt this idea to our own setting.

Definition 5.18. Let $\mathrm{EIL_{pb}}$ be given by

$$\phi ::= \mathtt{tt} \mid \neg\phi_c \mid \phi_c \wedge \phi'_c \mid \phi_r \wedge \phi_c \mid \phi_c \wedge \phi_r \mid \langle x : a\rangle\rangle\phi \mid \phi_r$$

where $\phi_r \in \mathrm{EIL_{dfro}}$ (without declarations $(x : a)\phi$), and ϕ_c ranges over closed formulas of $\mathrm{EIL_{pb}}$.

It can be seen that $\mathrm{EIL_{pb}}$ is obtained from forward moves $\langle x : a\rangle\rangle\phi$, reverse-only moves ϕ_r, and taking conjunctions of reverse-only and closed formulas, and negations and conjunctions of closed formulas. This logic is strong enough to encode pomset transitions.

Proposition 5.19. Let p be any pomset. There is a formula scheme $\langle p\rangle\rangle\phi$ such that for any closed formula $\phi \in \mathrm{EIL_{pb}}$:

— $\langle p\rangle\rangle\phi \in \mathrm{EIL_{pb}}$;

— for any configuration X of a stable configuration structure $\mathcal{C}$, we have $X \models \langle p\rangle\rangle\phi$ if and only if there is X' such that $X \xrightarrow{p}_{\mathcal{C}} X'$ and $X' \models \phi$.

Proof (sketch). Let $(X, <, \ell)$ be a representative of p, with $X = \{e_1, \ldots, e_n\}$ and $\ell(e_i) = a_i$ for each i, and with events ordered in such a way that if $e_i < e_j$, then $i < j$. Recall the open formula $\theta'_X \in \mathrm{EIL_{dfro}}$ from the proof of Lemma 5.4. There it was defined only for X a configuration, but it can be defined in the same way for any labelled poset. We define

$$\langle p\rangle\rangle\phi \stackrel{\mathrm{df}}{=} \langle z_1 : a_1\rangle\rangle \cdots \langle z_n : a_n\rangle\rangle(\theta'_X \wedge \phi).$$

Suppose Y is any configuration of a stable configuration structure $\mathcal{D}$. If $Y \models \langle p\rangle\rangle\phi$, then there are events $\{e'_1, \ldots, e'_n\}$ such that $\ell(e'_i) = a_i$ for each i, and $Y_1, \ldots, Y_n$ such that

$$Y \xrightarrow{e'_1}_{\mathcal{D}} Y_1 \cdots \xrightarrow{e'_n}_{\mathcal{D}} Y_n,$$

with

$$Y_n, \rho' \models \theta'_X \wedge \phi.$$

Here ρ' assigns z_i to e'_i for each i. $Y_n, \rho' \models \theta'_X$ tells us that $\{e'_1, \ldots, e'_n\}$ (with the ordering induced from Y_n) is isomorphic to X, so

$$Y \xrightarrow{p}_{\mathcal{D}} Y_n \models \phi$$

as required.

Conversely, if

$$Y \xrightarrow{p}_{\mathcal{D}} Y_n \models \phi,$$

https://doi.org/10.1017/S0960129513000510 Published online by Cambridge University Press

we list the members of $Y_n \setminus Y$ as $\{e'_1, \dots, e'_n\}$ in such a way that e'_i corresponds to e_i for each i. Then it is not hard to see that

$$Y \models \langle z_1 : a_1 \rangle\!\rangle \cdots \langle z_n : a_n \rangle\!\rangle (\theta'_X \wedge \phi),$$

where we assign each z_i to e'_i. Hence $Y \models \langle p \rangle\!\rangle \phi$ as required. □

In order to prove that $\mathrm{EIL}_{\mathrm{pb}}$ characterises PB, we will need some lemmas.

Lemma 5.20. Any formula of $\mathrm{EIL}_{\mathrm{pb}}$ is of one of the following two forms:

(1) $\phi_r \wedge \phi_c$.
(2) $\langle x : a \rangle\!\rangle \phi$ where $\langle x : a \rangle\!\rangle \phi$ is open.

Here we identify formulas up to commutativity and associativity of conjunction, and identify $\mathrm{tt} \wedge \phi$ with ϕ.

Proof. The result is trivial. □

The following lemma is similar to Lemma 5.3, but stated for $\mathrm{EIL}_{\mathrm{dfro}}$ rather than for $\mathrm{EIL}_{\mathrm{ro}}$ (just take $X = Y = \varnothing$ to recover Lemma 5.3 for $\mathrm{EIL}_{\mathrm{dfro}}$).

Lemma 5.21. Let $\mathcal{C}$ and $\mathcal{D}$ be stable configuration structures. Let X, X' be configurations of $\mathcal{C}$ with $X \subseteq X'$, and let Y, Y' be configurations of $\mathcal{D}$ with $Y \subseteq Y'$. Suppose

$$f : X' \setminus X \cong Y' \setminus Y.$$

Then for any $\phi \in \mathrm{EIL}_{\mathrm{dfro}}$, and any ρ (a permissible environment for ϕ and X) such that $\mathsf{rge}(\rho_\phi) \subseteq X' \setminus X$, we have $X', \rho \models \phi$ if and only if $Y', f \circ \rho_\phi \models \phi$.

Proof. The proof is by induction on ϕ. The cases for tt, negation and conjunction are as in the proof of Lemma 5.3, which just leaves the case for $\langle\!\langle x \rangle \phi$.

Suppose $X', \rho \models \langle\!\langle x \rangle \phi$ with

$$\mathsf{rge}(\rho_{\langle\!\langle x \rangle \phi}) \subseteq X' \setminus X.$$

Then $X' \overset{e}{\rightsquigarrow}_{\mathcal{C}} X''$ for some X'' and $X'', \rho \models \phi$. Now

$$\mathsf{rge}(\rho_\phi) \subseteq \mathsf{rge}(\rho_{\langle\!\langle x \rangle \phi}).$$

We also have

$$\mathsf{rge}(\rho_\phi) \subseteq X'',$$

since $X'', \rho \models \phi$. Combining these, we get

$$\mathsf{rge}(\rho_\phi) \subseteq X'' \setminus X.$$

We now let

$$\begin{aligned} e' &= f(e) \\ Y'' &= Y' \setminus \{e'\} \\ f' &= f \setminus \{(e, e')\}. \end{aligned}$$

Then $Y' \overset{e'}{\rightsquigarrow}_{\mathcal{D}} Y''$ and

$$f' : X'' \setminus X \cong Y'' \setminus Y.$$

https://doi.org/10.1017/S0960129513000510 Published online by Cambridge University Press

By the induction hypothesis,

$$Y'', f' \circ \rho_\phi \models \phi.$$

Since

$$f' \circ \rho_\phi = f \circ \rho_\phi,$$

we have

$$Y'', f \circ \rho_\phi \models \phi,$$

so

$$Y', f \circ \rho_\phi \models \langle\langle x \rangle\phi$$

as required.

Conversely, if $Y', f \circ \rho \models \langle\langle x \rangle\phi$, then $X', \rho \models \langle\langle x \rangle\phi$. □

Note that the induction in the proof of Lemma 5.21 would fail for declarations $(x : a)\phi$ since x might be assigned to an event outside $X' \setminus X$. This is why we stated Lemma 5.21 for $\mathrm{EIL}_{\mathrm{dfro}}$ rather than $\mathrm{EIL}_{\mathrm{ro}}$.

Theorem 5.22. Let $\mathcal{C}$ and $\mathcal{D}$ be stable configuration structures. Then $\mathcal{C} \approx_{\mathsf{pb}} \mathcal{D}$ if and only if $\mathcal{C} \sim_{\mathrm{EIL}_{\mathrm{pb}}} \mathcal{D}$.

Proof.

(⇒) Let $\mathcal{R}$ be a PB between $\mathcal{C}$ and $\mathcal{D}$. We shall show by induction on *closed* formulas that if $\mathcal{R}(X, Y)$, then $X \models \phi$ if and only if $Y \models \phi$:

— Cases $\phi = \mathrm{tt}$, $\phi = \neg\phi_c$ and $\phi = \phi_c \wedge \phi'_c$:
These cases are trivial.

— Case $\phi = \langle x_1 : a_1 \rangle\rangle\phi_1$:
For this case we shall use Lemma 5.20 repeatedly, starting with ϕ_1. Let n be such that

$$\phi_1 = \langle x_2 : a_2 \rangle\rangle\phi_2, \ldots, \phi_{n-1} = \langle x_n : a_n \rangle\rangle\phi_n$$

with $\phi_1, \ldots, \phi_{n-1}$ open and $\phi_n = \phi^n_r \wedge \phi^n_c$. Here n could of course be 1.

Suppose $X \models \phi$. There are events $e_1, \ldots, e_n$, configurations $X_1, \ldots, X_n$ and environments $\rho_1, \ldots, \rho_n$ such that

$$X \xrightarrow{e_1}_{\mathcal{C}} X_1 \cdots \xrightarrow{e_n}_{\mathcal{C}} X_n,$$

where $\ell(e_i) = a_i$ and $X_i, \rho_i \models \phi_i$ for $i = 1, \ldots, n$. Here ρ_i assigns $x_1, \ldots, x_i$ to $e_1, \ldots, e_i$ respectively.

Now let p be the pomset associated with the labelled partial order

$$(\{e_1, \ldots, e_n\}, <_X \restriction \{e_1, \ldots, e_n\}, \ell \restriction \{e_1, \ldots, e_n\}).$$

We have $X \xrightarrow{p}_{\mathcal{C}} X_n$. Hence there is Y_n such that $Y \xrightarrow{p}_{\mathcal{D}} Y_n$ and $\mathcal{R}(X_n, Y_n)$. Let

$$Y_n \setminus Y = \{e'_1, \ldots, e'_n\}$$

https://doi.org/10.1017/S0960129513000510 Published online by Cambridge University Press

with $f(e_i) = e'_i$ for $i = 1, \ldots, n$ being an order isomorphism between $\{e_1, \ldots, e_n\}$ and $\{e'_1, \ldots, e'_n\}$. Then

$$Y \xrightarrow{e'_1}_{\mathcal{D}} Y_1 \cdots \xrightarrow{e'_n}_{\mathcal{D}} Y_n$$

and $\ell(e_i) = a_i$ for $i = 1, \ldots, n$.

Now $X_n, \rho_n \models \phi_r^n$. By Lemma 5.21, we have $Y_n, f \circ \rho_n \models \phi_r^n$. Furthermore, $X_n \models \phi_c^n$. Hence, $Y_n \models \phi_c^n$ (using the induction hypothesis) so $Y \models \phi$ as required.

The converse is similar.

($\Leftarrow$) Suppose $\mathcal{C} \sim_{\mathrm{EIL_{pb}}} \mathcal{D}$. We define $\mathcal{R}$ by $\mathcal{R}(X, Y)$ if for all *closed* formulas $\phi \in \mathrm{EIL_{pb}}$, $X \models \phi$ if and only if $Y \models \phi$. We shall show that $\mathcal{R}$ is a pomset bisimulation.

It is clear that $\mathcal{R}(\emptyset, \emptyset)$ since $\mathcal{C} \sim_{\mathrm{EIL_{pb}}} \mathcal{D}$.

So we suppose $\mathcal{R}(X, Y)$, and further suppose that $X \xrightarrow{p}_{\mathcal{C}} X'$. Then $X \models \langle p \rangle\!\rangle \mathrm{tt}$, where $\langle p \rangle\!\rangle$ is as in Proposition 5.19. So $Y \models \langle p \rangle\!\rangle \mathrm{tt}$. Hence there is Y' such that $Y \xrightarrow{p}_{\mathcal{D}} Y'$. Let all such Y' be enumerated as Y_i ($i \in I$). We want to show that $\mathcal{R}(X', Y_i)$ for some i. In order to show a contradiction, we now suppose that for each i there is a closed formula $\phi_i \in \mathrm{EIL_{pb}}$ such that $X' \models \phi_i$ but $Y_i \not\models \phi_i$. Then $X \models \langle p \rangle\!\rangle \left(\bigwedge_{i \in I} \phi_i\right)$, but $Y \not\models \langle p \rangle\!\rangle \left(\bigwedge_{i \in I} \phi_i\right)$, which gives a contradiction. Hence $\mathcal{R}(X', Y_i)$ for some i.

Conversely, if $Y \xrightarrow{p}_{\mathcal{D}} Y'$, then $X \xrightarrow{p}_{\mathcal{C}} X'$ for some X'. □

If we allow reverse-only formulas in $\mathrm{EIL_{pb}}$ to contain declarations, we get a strictly stronger logic.

Definition 5.23. Let $\mathrm{EIL_{whpb}}$ be

$$\phi ::= \mathrm{tt} \mid \neg\phi_c \mid \phi_c \wedge \phi'_c \mid \phi_r \wedge \phi_c \mid \phi_c \wedge \phi_r \mid \langle x : a \rangle\!\rangle \phi \mid \phi_r$$

where $\phi_r \in \mathrm{EIL_{ro}}$ *with declarations* $(x : a)\phi$ and ϕ_c ranges over closed formulas of $\mathrm{EIL_{whpb}}$.

Thus $\mathrm{EIL_{whpb}}$ is obtained by adding declarations to reverse-only formulas in $\mathrm{EIL_{pb}}$. This logic is easily seen to include both $\mathrm{EIL_{wh}}$ and $\mathrm{EIL_{pb}}$.

Theorem 5.24. Let $\mathcal{C}$ and $\mathcal{D}$ be stable configuration structures. Then $\mathcal{C} \approx_{\mathsf{whpb}} \mathcal{D}$ if and only if $\mathcal{C} \sim_{\mathrm{EIL_{whpb}}} \mathcal{D}$.

Proof.

($\Rightarrow$) Let $\mathcal{R}$ be a WHPB between $\mathcal{C}$ and $\mathcal{D}$. We shall show by induction on *closed* formulas of $\mathrm{EIL_{whpb}}$ that if $\mathcal{R}(X, Y)$, then $X \models \phi$ if and only if $Y \models \phi$:

— Cases $\phi = \mathrm{tt}$, $\phi = \neg\phi_c$ and $\phi = \phi_c \wedge \phi'_c$:
These cases are trivial.

— Case $\phi = \langle x_1 : a_1 \rangle\!\rangle \phi_1$:
This case is much the same as the corresponding case in the proof of Theorem 5.22, noting that Lemma 5.20 would also hold for $\mathrm{EIL_{whpb}}$. The only difference is that we use Lemma 5.4 instead of Lemma 5.21 to deduce that

$$X_n, \rho_n \models \phi_r^n$$

https://doi.org/10.1017/S0960129513000510 Published online by Cambridge University Press

implies

$$Y_n, f \circ \rho_n \models \phi_r^n$$

(we know that $X_n \cong Y_n$ since $\mathcal{R}$ is a WHPB).

— Case $\phi = \phi_r \in \mathrm{EIL_{ro}}$:
This case follows from Lemma 5.3 since $X \cong Y$.

($\Leftarrow$) Suppose $\mathcal{C} \sim_{\mathrm{EIL_{whpb}}} \mathcal{D}$. We define $\mathcal{R}$ by $\mathcal{R}(X, Y)$ if for all *closed* formulas $\phi \in \mathrm{EIL_{whpb}}$, $X \models \phi$ if and only if $Y \models \phi$. The closed formulas of $\mathrm{EIL_{whpb}}$ include all of $\mathrm{EIL_{wh}}$ and the closed formulas of $\mathrm{EIL_{pb}}$. Therefore, by the proofs of Theorems 5.15 and 5.22, $\mathcal{R}$ is both a weak history-preserving and a pomset bisimulation. Hence $\mathcal{R}$ is a WHPB and $\mathcal{C} \approx_{\mathsf{whpb}} \mathcal{D}$. □

We conclude by noting that logics for SB and IB can be defined straightforwardly. Let the logic $\mathrm{EIL_{sb}}$ be given by

$$\phi ::= \mathrm{tt} \mid \neg\phi \mid \phi \wedge \phi' \mid \langle A \rangle\rangle \phi$$

(all multisets A). Note that all formulas are closed. It is easy to see that it is a sublogic of $\mathrm{EIL_{pb}}$. The logic $\mathrm{EIL_{sb}}$ is very similar to the corresponding logic for step bisimulation given by Baldan and Crafa (2010, Theorem 2). It is straightforward to show that $\mathcal{C} \approx_{\mathsf{sb}} \mathcal{D}$ if and only if $\mathcal{C} \sim_{\mathrm{EIL_{sb}}} \mathcal{D}$ – see the proof given in Baldan and Crafa (2011, Theorem 2).

Remark 5.25. The logics we have found generally mirror the inclusions in Figure 1 in that whenever an inclusion holds, the corresponding logics are included in each other (in the opposite direction). However, there is one exception: $\approx_{\mathsf{wh}} \subsetneq \approx_{\mathsf{sb}}$ but $\mathrm{EIL_{sb}}$ is not a sublogic of $\mathrm{EIL_{wh}}$ (as a simple example, $\langle a, b \rangle\rangle \mathrm{tt}$ is a formula of $\mathrm{EIL_{sb}}$ but not of $\mathrm{EIL_{wh}}$). This is not all that surprising since the inclusion $\approx_{\mathsf{wh}} \subsetneq \approx_{\mathsf{sb}}$ is non-obvious. It would be of interest to find alternative logics for SB and WH such that the logic for $\approx_{\mathsf{sb}}$ is a sublogic of the one for $\approx_{\mathsf{wh}}$. Of course, we could trivially solve this by taking the union of $\mathrm{EIL_{sb}}$ and $\mathrm{EIL_{wh}}$ as a logic for $\approx_{\mathsf{wh}}$, but we would like a more interesting and elegant solution.

Finally, let $\mathrm{EIL_{ib}}$ be given by

$$\phi ::= \mathrm{tt} \mid \neg\phi \mid \phi \wedge \phi' \mid \langle a \rangle\rangle \phi.$$

It is easy to see that this is a sublogic of $\mathrm{EIL_{sb}}$. Alternatively, $\mathrm{EIL_{ib}}$ can be obtained by taking full EIL (with declarations) and omitting all reverse moves. In this case, of course, the identifiers and declarations no longer add any power. We have $\mathcal{C} \approx_{\mathsf{ib}} \mathcal{D}$ if and only if $\mathcal{C} \sim_{\mathrm{EIL_{ib}}} \mathcal{D}$, which is, of course, simply the classical result for standard Hennessy–Milner logic.

6. Characteristic formulas

In this section we shall investigate characteristic formulas for three of the equivalences we have considered, namely, HH, H and WH. The idea is that we reduce checking whether $\mathcal{C}$ and $\mathcal{D}$ satisfy the same formulas in a logic such as EIL to the question of whether $\mathcal{D}$

https://doi.org/10.1017/S0960129513000510 Published online by Cambridge University Press

satisfies a particular formula $\chi_{\mathcal{C}}$, the *characteristic formula* of $\mathcal{C}$, which completely expresses the behaviour of $\mathcal{C}$, at least as far as the particular logic is concerned. As pointed out in Aceto *et al.* (2009), this means that checking whether two structures are equivalent is changed from the problem of potentially having to check infinitely many formulas into a single model-checking problem $\mathcal{D} \models \chi_{\mathcal{C}}$.

Characteristic formulas for models of concurrent systems were first investigated in Graf and Sifakis (1986), and subsequently in Steffen and Ingólfsdóttir (1994) and other papers – see Aceto *et al.* (2009) for further references. As far as we are aware, characteristic formulas have not previously been investigated for any true concurrency logic, although we should mention that Aceto *et al.* (2009) did study characteristic formulas for a logic with both forward and reverse modalities, which is related to the back and forth simulation of De Nicola *et al.* (1990).

We shall confine ourselves to *finite* stable configuration structures in this section. Even with this assumption, it is not obvious that an equivalence such as HH, which employs both forward and reverse transitions, can be captured by a single finite-depth formula. To show that forward and reverse transitions need not alternate for ever, we first relate HH to a simple game.

Definition 6.1. Let $\mathcal{C}$ and $\mathcal{D}$ be finite stable configuration structures. The game $G(\mathcal{C},\mathcal{D})$ has two players: Attacker and Defender. The set of game states is

$$S(\mathcal{C},\mathcal{D}) \stackrel{\text{df}}{=} \{(X,Y,f) : X \in C_{\mathcal{C}}, Y \in C_{\mathcal{D}}, f : X \cong Y\}.$$

The start state is $(\emptyset,\emptyset,\emptyset)$. At each state of the game, Attacker chooses a forward (respectively, reverse) move e of either $\mathcal{C}$ or $\mathcal{D}$. Then D must reply with a corresponding forward (respectively, reverse) move e' by the other structure. Going forwards, we extend f to f', and going in reverse, we restrict f to f', as in the definition of HH. The two moves produce a new game state (X',Y',f'). Defender wins if we get to a previously visited state, and Attacker wins if Defender cannot find a move (Defender also wins if Attacker cannot find a move, but that can only happen if both $\mathcal{C}$ and $\mathcal{D}$ only have the empty configuration).

It is reasonable for Defender to win if a state is repeated since if Attacker then chooses a different and better move at the repeated state, Attacker could have chosen that one on the previous occasion.

Definition 6.2. Given finite stable configuration structures $\mathcal{C}$ and $\mathcal{D}$, let

$$\begin{aligned} s(\mathcal{C},\mathcal{D}) &\stackrel{\text{df}}{=} |S(\mathcal{C},\mathcal{D})| \\ c(\mathcal{C}) &\stackrel{\text{df}}{=} \max\{|X| : X \in C_{\mathcal{C}}\} \\ c(\mathcal{C},\mathcal{D}) &\stackrel{\text{df}}{=} \min\{c(\mathcal{C}), c(\mathcal{D})\}. \end{aligned}$$

It is clear that any play of the game $G(\mathcal{C},\mathcal{D})$ finishes after no more than $s(\mathcal{C},\mathcal{D})$ moves. We can place an upper bound on $s(\mathcal{C},\mathcal{D})$ as follows.

https://doi.org/10.1017/S0960129513000510 Published online by Cambridge University Press

Proposition 6.3. Let $\mathcal{C}$ and $\mathcal{D}$ be finite stable configuration structures. Then

$$s(\mathcal{C},\mathcal{D}) \leqslant |C_{\mathcal{C}}|.|C_{\mathcal{D}}|.c(\mathcal{C},\mathcal{D})!.$$

Note that if there is no autoconcurrency, any isomorphism $f : X \cong Y$ is unique, so we can improve the upper bound on the number of states to

$$s(\mathcal{C},\mathcal{D}) \leqslant |C_{\mathcal{C}}|.|C_{\mathcal{D}}|.$$

Proposition 6.4. Let $\mathcal{C}$ and $\mathcal{D}$ be finite stable configuration structures. Then $\mathcal{C} \approx_{\mathsf{hh}} \mathcal{D}$ if and only if Defender has a winning strategy for the game $G(\mathcal{C},\mathcal{D})$.

Proof (sketch).

($\Rightarrow$) Suppose $\mathcal{R}$ is an HH bisimulation between $\mathcal{C}$ and $\mathcal{D}$. Note that $\mathcal{R}(\emptyset,\emptyset,\emptyset)$, so that the initial state of $G(\mathcal{C},\mathcal{D})$ is in $\mathcal{R}$. Hence Defender has a winning strategy as follows. Always choose a move that produces a new state (X',Y',f') so that $\mathcal{R}(X',Y',f')$. This is clearly possible by the properties of $\mathcal{R}$. Since Defender is always able to make a move, a state will be repeated eventually since there are only finitely many possible states (X,Y,f). In fact, there can be no more than $s(\mathcal{C},\mathcal{D})$ moves before Defender wins, as already observed.

($\Leftarrow$) Suppose Defender has a winning strategy for $G(\mathcal{C},\mathcal{D})$. We define $\mathcal{R}(X,Y,f)$ if and only if (X,Y,f) is reachable in some play of $G(\mathcal{C},\mathcal{D})$ (where we assume that Defender always plays their winning strategy). It is clear that $\mathcal{R}(\emptyset,\emptyset,\emptyset)$. Also, if $\mathcal{R}(X,Y,f)$, then any transition of $\mathcal{C}$ or $\mathcal{D}$ can be matched (since Defender has a winning strategy), so we can get to a new reachable state (X',Y',f'), and thus $\mathcal{R}(X',Y',f')$ as required. The only exception is if we have reached a winning (for Defender) state (X,Y,f), but in that case this same state was reached earlier in the play, so we can use the earlier occurrence instead. □

Remark 6.5. Game characterisations of HH equivalence have been used many times before – see, for example, Fröschle (1999), Fröschle (2005), Fröschle and Lasota (2005), Jurdzinski *et al.* (2003) and Gutierrez (2009). However, Defender is usually said to win if the play continues for ever, whereas we say that Defender wins if a state is repeated. This is because we are working with finite configuration structures, rather than, say, Petri nets.

Definition 6.6. Let $\phi \in$ EIL. The *modal depth* $\mathsf{md}(\phi)$ of ϕ is defined as follows:

$$\begin{aligned}
\mathsf{md}(\mathsf{tt}) &\stackrel{\mathrm{df}}{=} 0\\
\mathsf{md}(\neg\phi) &\stackrel{\mathrm{df}}{=} \mathsf{md}(\phi)\\
\mathsf{md}(\phi \wedge \phi') &\stackrel{\mathrm{df}}{=} \max(\mathsf{md}(\phi),\mathsf{md}(\phi'))\\
\mathsf{md}(\langle x : a\rangle\rangle\phi) &\stackrel{\mathrm{df}}{=} 1 + \mathsf{md}(\phi)\\
\mathsf{md}((x : a)\phi) &\stackrel{\mathrm{df}}{=} \mathsf{md}(\phi)\\
\mathsf{md}(\langle\langle x : a\rangle\phi) &\stackrel{\mathrm{df}}{=} 1 + \mathsf{md}(\phi).
\end{aligned}$$

https://doi.org/10.1017/S0960129513000510 Published online by Cambridge University Press

We can use the game characterisation of HH to bound the modal depth of the EIL formulas needed to check whether finite structures are HH equivalent.

Theorem 6.7. Let $\mathcal{C}$ and $\mathcal{D}$ be finite stable configuration structures. Then $\mathcal{C} \approx_{\mathsf{hh}} \mathcal{D}$ if and only if $\mathcal{C}$ and $\mathcal{D}$ satisfy the same EIL formulas of modal depth no more than

$$s(\mathcal{C}, \mathcal{D}) + c(\mathcal{C}, \mathcal{D}).$$

Proof.

($\Rightarrow$) This direction follows immediately from Theorem 5.9.

($\Leftarrow$) Let $s = s(\mathcal{C}, \mathcal{D})$ and $c = c(\mathcal{C}, \mathcal{D})$. Let EIL^k be those formulas of EIL with modal depth $\leqslant k$. Suppose $\mathcal{C}$ and $\mathcal{D}$ satisfy the same EIL^{s+c} formulas. We aim to show that Defender has a winning strategy for $G(\mathcal{C}, \mathcal{D})$.

The game starts in stage 0 and goes through stages 1 up to no more than s. We shall show by induction on k that Defender has a winning strategy where at stage k, in state (X, Y, f) with $f : X \cong Y$, it is the case that for any $\phi \in \mathrm{EIL}^{s+c-k}$ and any ρ (a permissible environment for ϕ and X) with $\mathsf{rge}(\rho) \subseteq X$, we have $X, \rho \models \phi$ if and only if $Y, f \circ \rho \models \phi$:

— Base case $k = 0$:
This follows immediately from the assumption that $\mathcal{C}$ and $\mathcal{D}$ satisfy the same EIL^{s+c} formulas.

— Induction step:
Suppose that at stage k (where $k \leqslant s - 1$) we are in state (X, Y, f), and suppose that for any $\phi \in \mathrm{EIL}^{s+c-k}$ and any ρ (a permissible environment for ϕ and X) with $\mathsf{rge}(\rho) \subseteq X$ we have $X, \rho \models \phi$ if and only if $Y, f \circ \rho \models \phi$.

We must now show that whatever move Attacker makes, Defender can respond in such a way as to get to a new state (X', Y', f') where $f' : X' \cong Y'$ and for any $\phi \in \mathrm{EIL}^{s+c-k-1}$ and any ρ' (a permissible environment for ϕ and X') with $\mathsf{rge}(\rho') \subseteq X'$, we have $X', \rho' \models \phi$ if and only if $Y', f' \circ \rho' \models \phi$. We consider cases:

– Attacker plays $X \xrightarrow{e}_{\mathcal{C}} X'$:
Then Defender must respond with $Y \xrightarrow{e'}_{\mathcal{D}} Y'$ such that $f' : X' \cong Y'$ where $f' = f \cup \{(e, e')\}$ and for any $\phi \in \mathrm{EIL}^{s+c-k-1}$ and any ρ' (a permissible environment for ϕ and X') with $\mathsf{rge}(\rho') \subseteq X'$, we have

$$X', \rho' \models \phi$$

if and only if

$$Y', f' \circ \rho' \models \phi.$$

To see that Defender does have such a move, we follow the corresponding case in the proof of Theorem 5.9. Note that

$$\mathsf{md}(\theta'_{X'}) \leqslant |X'| \leqslant c$$

and that the ϕ_i are bounded in modal depth by $s + c - k - 1$. Hence, the ψ_i are bounded in modal depth by $s + c - k - 1$ since $k \leqslant s - 1$, so $c \leqslant$

https://doi.org/10.1017/S0960129513000510 Published online by Cambridge University Press

$s + c - k - 1$. Therefore, $\langle z_e : a \rangle\rangle \Psi \in \mathrm{EIL}^{s+c-k}$, allowing us to obtain the contradiction required.

- Attacker plays $Y \xrightarrow{e'}_{\mathcal{D}} Y'$:
  This is similar to the previous case.
- Attacker plays $X \overset{e}{\rightsquigarrow}_{\mathcal{C}} X'$:
  Then Defender must respond with $Y \overset{f(e)}{\rightsquigarrow}_{\mathcal{D}} Y'$ such that $f' : X' \cong Y'$ where $f' = f \upharpoonright X'$ and for any $\phi \in \mathrm{EIL}^{s+c-k-1}$ and any ρ' (a permissible environment for ϕ and X') with $\mathsf{rge}(\rho') \subseteq X'$, we have

  $$X', \rho' \models \phi$$

  if and only if

  $$Y', f' \circ \rho' \models \phi.$$

  To see that Defender does have such a move, we follow the corresponding case in the proof of Theorem 5.9. Note that

  $$\phi \in \mathrm{EIL}^{s+c-k-1},$$

  so

  $$\langle\langle z \rangle \phi \in \mathrm{EIL}^{s+c-k},$$

  allowing us to obtain the contradiction required.
- Attacker plays $Y \overset{e'}{\rightsquigarrow}_{\mathcal{D}} Y'$:
  This is similar to the previous case. □

We now define a family of characteristic formulas $\chi^{\mathrm{hh}}_{X,n}$ for HH equivalence parametrised on modal depth n and defined by mutual recursion on the configurations X of a configuration structure $\mathcal{C}$. The formula $\chi^{\mathrm{hh}}_{X,n+1}$ will be the conjunction of:

— a formula giving the order isomorphism class of X (which is possible by Lemma 5.7);

— a formula stating that for any forward transition $X \xrightarrow{e}_{\mathcal{C}} X'$, it is possible to perform an event labelled with $\ell(e)$ and reach a state where $\chi^{\mathrm{hh}}_{X',n}$ holds (note that the depth parameter decreases, so this is a well-defined recursion);

— a formula stating that for any label a, performing any event labelled with a takes us to a state where $\chi^{\mathrm{hh}}_{X',n}$ holds for some some X' such that $X \xrightarrow{a}_{\mathcal{C}} X'$;

— a formula stating that for any reverse transition $X \overset{e}{\rightsquigarrow}_{\mathcal{C}} X'$, it is possible to perform an event labelled with $\ell(e)$ and reach a state where $\chi^{\mathrm{hh}}_{X',n}$ holds.

Thus the various conjuncts correspond to the definition of HH bisimulation.

https://doi.org/10.1017/S0960129513000510 Published online by Cambridge University Press

Definition 6.8. Suppose Act is finite. Let $\mathcal{C}$ be a finite stable configuration structure. We define formulas $\chi^{\mathrm{hh}}_{X,n}$ (X a configuration of $\mathcal{C}$) by induction on n as follows:

$$\begin{aligned}\chi^{\mathrm{hh}}_{X,0} &\stackrel{\mathrm{df}}{=} \theta'_X \\ \chi^{\mathrm{hh}}_{X,n+1} &\stackrel{\mathrm{df}}{=} \theta'_X \wedge \left(\bigwedge_{X \xrightarrow{e}_{\mathcal{C}} X'} \langle z_e : \ell(e) \rangle\!\rangle \chi^{\mathrm{hh}}_{X',n} \right) \\ &\quad \wedge \left(\bigwedge_{a \in \mathsf{Act}} [x : a]] \bigvee_{X \xrightarrow{e}_{\mathcal{C}} X', \ell(e)=a} \chi^{\mathrm{hh}}_{X',n}[x/z_e] \right) \wedge \left(\bigwedge_{X \stackrel{e}{\rightsquigarrow}_{\mathcal{C}} X'} \langle\!\langle z_e \rangle \chi^{\mathrm{hh}}_{X',n} \right)\end{aligned}$$

Here $\theta'_X \in \mathrm{EIL}_{\mathrm{dfro}}$ is as in Lemma 5.7 and

$$\mathsf{fi}(\chi^{\mathrm{hh}}_{X,n}) = \{z_e : e \in X\}.$$

We further let

$$\chi^{\mathrm{hh}}_{\mathcal{C},n} \stackrel{\mathrm{df}}{=} \chi^{\mathrm{hh}}_{\emptyset,n}.$$

Note that

$$\chi^{\mathrm{hh}}_{X,n} \in \mathrm{EIL}$$

and

$$\mathsf{md}(\chi^{\mathrm{hh}}_{X,n}) \leqslant n + c(\mathcal{C}).$$

Theorem 6.9. Suppose Act is finite. Let $\mathcal{C}$ and $\mathcal{D}$ be finite stable configuration structures. Let $s \stackrel{\mathrm{df}}{=} s(\mathcal{C},\mathcal{D})$. Then $\mathcal{C} \approx_{\mathsf{hh}} \mathcal{D}$ if and only if $\mathcal{D} \models \chi^{\mathrm{hh}}_{\mathcal{C},s}$.

Proof.

($\Rightarrow$) It is easy to see by induction on n that $\mathcal{C} \models \chi^{\mathrm{hh}}_{\mathcal{C},n}$ for any n. Now suppose $\mathcal{C} \approx_{\mathsf{hh}} \mathcal{D}$. Then $\mathcal{C} \models \chi^{\mathrm{hh}}_{\mathcal{C},s}$, so $\mathcal{D} \models \chi^{\mathrm{hh}}_{\mathcal{C},s}$ by Theorem 5.9.

($\Leftarrow$) We show that Defender has a strategy to win the game $G(\mathcal{C},\mathcal{D})$.

Let ρ_X be defined by $\rho_X(e) = z_e$ for each $e \in X$. Defender must ensure that at each stage $k \leqslant s$ in state (X, Y, f), we have

$$Y, f \circ \rho_X \models \chi^{\mathrm{hh}}_{X,s-k}.$$

This is true initially at $k = 0$ in state $(\emptyset, \emptyset, \emptyset)$ since $\mathcal{D} \models \chi^{\mathrm{hh}}_{\mathcal{C},s}$.

At each stage $k < s$, Defender must choose a response that ensures that

$$Y', f' \circ \rho_{X'} \models \chi^{\mathrm{hh}}_{X',s-k-1},$$

where (X', Y', f') is the new state.

https://doi.org/10.1017/S0960129513000510 Published online by Cambridge University Press

Considering cases:

— Attacker plays $X \xrightarrow{e} X'$:
We know

$$Y, f \circ \rho_X \models \langle z_e : \ell(e)\rangle\rangle \chi^{\mathrm{hh}}_{X',s-k-1},$$

so $Y \xrightarrow{e'}_{\mathcal{D}} Y'$ where

$$Y', (f \circ \rho_X)[z_e \mapsto e'] \models \chi^{\mathrm{hh}}_{X',s-k-1}.$$

Let $f' = f \cup \{(e, e')\}$. Then

$$Y', f' \circ \rho_{X'} \models \chi^{\mathrm{hh}}_{X',s-k-1}.$$

In particular,

$$Y', f' \circ \rho_{X'} \models \theta'_{X'}.$$

Hence $f' : X' \cong Y'$ by Lemma 5.7 (or the proof of Lemma 5.4). So Defender has found a valid move and maintained the induction hypothesis.

— Attacker plays $Y \xrightarrow{e'}_{\mathcal{D}} Y'$:
Let $\ell(e') = a$. We know

$$Y, f \circ \rho_X \models [x : a]] \bigvee_{X \xrightarrow{e}_{\mathcal{C}} X', \ell(e)=a} \chi^{\mathrm{hh}}_{X',s-k-1}[x/z_e],$$

so

$$Y', (f \circ \rho_X)[x \mapsto e'] \models \bigvee_{X \xrightarrow{e}_{\mathcal{C}} X', \ell(e)=a} \chi^{\mathrm{hh}}_{X',s-k-1}[x/z_e].$$

This disjunction cannot be empty since otherwise

$$Y', (f \circ \rho_X)[x \mapsto e'] \models \mathrm{ff},$$

which is impossible. So there is e such that $X \xrightarrow{e}_{\mathcal{C}} X'$, $\ell(e) = a$ and

$$Y', (f \circ \rho_X)[x \mapsto e'] \models \chi^{\mathrm{hh}}_{X',s-k-1}[x/z_e].$$

So Defender plays $X \xrightarrow{e}_{\mathcal{C}} X'$. Let $f' = f \cup \{(e, e')\}$. Then

$$Y', f' \circ \rho_{X'} \models \chi^{\mathrm{hh}}_{X',s-k-1}.$$

Hence $f' : X' \cong Y'$ just as in the previous case and Defender has again found a valid move and maintained the induction hypothesis.

— Attacker plays $X \overset{e}{\rightsquigarrow}_{\mathcal{C}} X'$:
Then Defender plays $Y \xrightarrow{f(e)}_{\mathcal{D}} Y'$. Let $f' = f \restriction X'$. We know

$$Y, f \circ \rho_X \models \langle\langle z_e \rangle \chi^{\mathrm{hh}}_{X',s-k-1},$$

so

$$Y', f' \circ \rho_{X'} \models \chi^{\mathrm{hh}}_{X',s-k-1}.$$

Hence $f' : X' \cong Y'$ just as in the previous cases and Defender has again found a valid move and maintained the induction hypothesis.

https://doi.org/10.1017/S0960129513000510 Published online by Cambridge University Press

— Attacker plays $Y \stackrel{e'}{\rightsquigarrow}_{\mathcal{D}} Y'$:
Let $e = f^{-1}(e')$. Then Defender plays $X \stackrel{e}{\rightsquigarrow}_{\mathcal{C}} X'$. Let $f' = f \restriction X'$. We again know

$$Y, f \circ \rho_X \models \langle\!\langle z_e \rangle\chi^{\mathrm{hh}}_{X',s-k-1},$$

so again

$$Y', f' \circ \rho_{X'} \models \chi^{\mathrm{hh}}_{X',s-k-1}.$$

Hence $f' : X' \cong Y'$ just as in the previous cases and Defender has again found a valid move and maintained the induction hypothesis. □

Theorem 6.9 does not give us a single characteristic formula for $\mathcal{C}$, but it does allow us to deal uniformly with all $\mathcal{D}$s up to a certain size. This is almost as good as having a single characteristic formula for $\mathcal{C}$ since we can generate a formula of the appropriate size once we have settled on $\mathcal{D}$, so we have still reduced equivalence checking to checking a single formula. Single characteristic formulas are certainly possible for some $\mathcal{C}$s (see Example 6.10 below), but whether there is a single formula $\chi^{\mathrm{hh}}_{\mathcal{C}}$ for all finite $\mathcal{C}$ that works for all $\mathcal{D}$ remains an open question.

Example 6.10. Consider the configuration structure represented by the CCS process a, which we denote by $\mathcal{C}_a$. This has configurations $\emptyset$ and $\{e\}$ with $\ell(e) = a$. The single formula

$$\phi_a \stackrel{\mathrm{df}}{=} \langle x : a \rangle\!\rangle \mathrm{tt} \wedge \left([x : a]] \bigwedge_{b \in \mathsf{Act}} [y : b]]\, \mathrm{ff} \right) \wedge \bigwedge_{b \in \mathsf{Act}, b \neq a} [y : b]]\, \mathrm{ff}$$

characterises $\mathcal{C}_a$ for HH equivalence, as we shall now show. It is clear that $\mathcal{C}_a$ satisfies ϕ_a. We claim that for any structure $\mathcal{C}$, if $\mathcal{C}$ satisfies ϕ_a, then $\mathcal{C} \approx_{\mathsf{hh}} \mathcal{C}_a$. So we suppose $\mathcal{C}$ satisfies ϕ_a. It is clear that any single-event configuration of $\mathcal{C}$ must be labelled with a. If $\mathcal{C}$ had a configuration Y with two elements, we would have $\emptyset \stackrel{a}{\rightarrow}_{\mathcal{C}} X \stackrel{b}{\rightarrow}_{\mathcal{C}} Y$ for some single-event X and some b. But this is not possible by the second conjunct of ϕ_a. So any configuration of $\mathcal{C}$ is either the empty set or of the form $\{e'\}$ for some e' with $\ell(e') = a$. It is now easy to define an HH bisimulation between $\mathcal{C}_a$ and $\mathcal{C}$.

We can generalise this example in two ways:

(1) Consider the configuration structure $\mathcal{C}_s$ represented by a summation $s = a_1 + \cdots + a_n$ (where the a_i are not necessarily distinct). Let

$$A = \{a_i : i = 1, \ldots, n\}$$

(as a set rather than a multiset). The formula

$$\phi_s \stackrel{\mathrm{df}}{=} \bigwedge_{i=1}^{n} \left(\langle x : a_i \rangle\!\rangle \mathrm{tt} \wedge [x : a_i]] \bigwedge_{b \in \mathsf{Act}} [y : b]]\, \mathrm{ff} \right) \wedge \bigwedge_{b \in \mathsf{Act} \setminus A} [y : b]]\, \mathrm{ff}$$

is satisfied by $\mathcal{C}_s$. Also, if any $\mathcal{C}$ satisfies ϕ_s, then, much as above, any configuration is either empty or a single event with a label in A. Also, for $a \in A$, by the first conjunct, $\mathcal{C}$ must have a configuration $\{e\}$ with $\ell(e) = a$. It is now again straightforward to define an HH bisimulation between $\mathcal{C}_s$ and $\mathcal{C}$.

https://doi.org/10.1017/S0960129513000510 Published online by Cambridge University Press

(2) A second generalisation is to the configuration structure represented by a sequential chain $a_1.a_2 \dots a_n$ (where again the a_i are not necessarily distinct). A single formula that characterises this structure with respect to HH equivalence is ϕ_0 where

$$\phi_i \stackrel{\mathrm{df}}{=} \langle x_{i+1} : a_{i+1} \rangle\rangle \mathrm{tt} \wedge ([x_{i+1} : a_{i+1}]] \, \phi_{i+1}) \wedge \left(\bigwedge_{b \in \mathsf{Act}, b \neq a_{i+1}} [y : b]] \, \mathrm{ff} \right) \wedge \bigwedge_{j=1}^{i-1} [[x_j] \, \mathrm{ff}$$

for $i = 0, \dots, n-1$, and

$$\phi_n \stackrel{\mathrm{df}}{=} \left(\bigwedge_{b \in \mathsf{Act}} [y : b]] \, \mathrm{ff} \right) \wedge \bigwedge_{j=1}^{n-1} [[x_j] \, \mathrm{ff}.$$

We shall omit the checks, which are a generalisation of those for the $n = 1$ case already covered.

Matters are simpler for H and WH equivalences since then only forward transitions are employed.

Definition 6.11. Suppose Act is finite and let $\mathcal{C}$ be a finite stable configuration structure. We define formulas χ_X^{h} (X a configuration of $\mathcal{C}$) as follows:

$$\chi_X^{\mathrm{h}} \stackrel{\mathrm{df}}{=} \theta'_X \wedge \left(\bigwedge_{X \xrightarrow{e}_{\mathcal{C}} X'} \langle z_e : \ell(e) \rangle\rangle \chi_{X'}^{\mathrm{h}} \right) \wedge \left(\bigwedge_{a \in \mathsf{Act}} [x : a]] \bigvee_{X \xrightarrow{e}_{\mathcal{C}} X', \ell(e) = a} \chi_{X'}^{\mathrm{h}}[x/z_e] \right).$$

Here $\theta'_X \in \mathrm{EIL}_{\mathrm{dfro}}$ is as in Lemma 5.7. We further let $\chi_{\mathcal{C}}^{\mathrm{h}} \stackrel{\mathrm{df}}{=} \chi_{\emptyset}^{\mathrm{h}}$.

Note that $\chi_{\mathcal{C}}^{\mathrm{h}} \in \mathrm{EIL}_{\mathrm{h}}$, and $\chi_{\mathcal{C}}^{\mathrm{h}}$ is well defined since maximal configurations form the base cases of the recursion. Also $\mathsf{md}(\chi_X^{\mathrm{h}}) \leqslant 2.c(\mathcal{C})$.

Proposition 6.12. Suppose Act is finite and let $\mathcal{C}$ and $\mathcal{D}$ be finite stable configuration structures. Then $\mathcal{D} \approx_{\mathsf{h}} \mathcal{C}$ if and only if $\mathcal{D} \models \chi_{\mathcal{C}}^{\mathrm{h}}$.

Proof (sketch).

($\Rightarrow$) We shall first show that $\mathcal{C} \models \chi_{\mathcal{C}}^{\mathrm{h}}$. Let ρ_X be defined by $\rho_X(e) = z_e$ (each $e \in X$). We shall show that $X, \rho_X \models \chi_X^{\mathrm{h}}$ for each configuration X of $\mathcal{C}$. The proof is by induction on the maximum number of transitions from the current configuration to a maximal configuration:

$$d(X) \stackrel{\mathrm{df}}{=} \begin{cases} 0 & \text{if } X \text{ is maximal} \\ \max\{d(X') + 1 : X \xrightarrow{e} X'\} & \text{otherwise.} \end{cases}$$

We shall omit the straightforward details.

Now suppose $\mathcal{C} \approx_{\mathsf{h}} \mathcal{D}$. Then $\mathcal{C} \models \chi_{\mathcal{C}}^{\mathrm{h}}$, so $\mathcal{D} \models \chi_{\mathcal{C}}^{\mathrm{h}}$ by Theorem 5.12.

($\Leftarrow$) Suppose $\mathcal{D} \models \chi_{\mathcal{C}}^{\mathrm{h}}$. Let $\mathcal{R}(X, Y, f)$ if and only if $f : X \cong Y$ and $Y, f \circ \rho_X \models \chi_X^{\mathrm{h}}$. We shall show that $\mathcal{R}$ is an H-bisimulation between $\mathcal{C}$ and $\mathcal{D}$.

It is clear that $Y, f \circ \rho_X \models \chi_X^{\mathrm{h}}$ holds if $X = Y = \emptyset$, since $\mathcal{D} \models \chi_{\mathcal{C}}^{\mathrm{h}}$. Hence $\mathcal{R}(\emptyset, \emptyset, \emptyset)$.

https://doi.org/10.1017/S0960129513000510 Published online by Cambridge University Press

So we suppose $\mathcal{R}(X, Y)$ and consider cases:

— Case $X \xrightarrow{e}_{\mathcal{C}} X'$:
We know

$$Y, f \circ \rho_X \models \langle z_e : \ell(e) \rangle\!\rangle \chi^{\mathrm{h}}_{X'},$$

so $Y \xrightarrow{e'}_{\mathcal{D}} Y'$ where

$$Y', (f \circ \rho_X)[z_e \mapsto e'] \models \chi^{\mathrm{h}}_{X'}.$$

Let $f' = f \cup \{(e, e')\}$. Then

$$Y', f' \circ \rho_{X'} \models \chi^{\mathrm{h}}_{X'}.$$

In particular,

$$Y', f' \circ \rho_{X'} \models \theta'_{X'}.$$

Hence $f' : X' \cong Y'$ by Lemma 5.7 (or the proof of Lemma 5.4), and $\mathcal{R}(X', Y')$ as required.

— Case $Y \xrightarrow{e'}_{\mathcal{C}} Y'$:
Let $\ell(e') = a$. We know

$$Y, f \circ \rho_X \models [x : a]] \bigvee_{X \xrightarrow{e}_{\mathcal{C}} X', \ell(e)=a} \chi^{\mathrm{h}}_{X'}[x/z_e],$$

so

$$Y', (f \circ \rho_X)[x \mapsto e'] \models \bigvee_{X \xrightarrow{e}_{\mathcal{C}} X', \ell(e)=a} \chi^{\mathrm{h}}_{X'}[x/z_e].$$

This disjunction cannot be empty since otherwise

$$Y', (f \circ \rho_X)[x \mapsto e'] \models \mathrm{ff},$$

which is impossible. So there is e such that

$$X \xrightarrow{e}_{\mathcal{C}} X'$$
$$\ell(e) = a$$
$$Y', (f \circ \rho_X)[x \mapsto e'] \models \chi^{\mathrm{h}}_{X'}[x/z_e].$$

Let

$$f' = f \cup \{(e, e')\}.$$

Then

$$Y', f' \circ \rho_{X'} \models \chi^{\mathrm{h}}_{X'}.$$

Hence $f' : X' \cong Y'$ just as in the previous case, and $\mathcal{R}(X', Y')$ as required. □

WH is even easier since formulas are closed.

https://doi.org/10.1017/S0960129513000510 Published online by Cambridge University Press

Definition 6.13. Suppose Act is finite. Let $\mathcal{C}$ be a finite stable configuration structure. We define formulas χ_X^{wh} (X a configuration of $\mathcal{C}$) by

$$\chi_X^{\mathrm{wh}} \stackrel{\mathrm{df}}{=} \theta_X \ \wedge \ \left(\bigwedge_{X \xrightarrow{a}_{\mathcal{C}} X'} \langle a \rangle\rangle \chi_{X'}^{\mathrm{wh}} \right) \ \wedge \ \left(\bigwedge_{a \in \mathsf{Act}} [a]] \bigvee_{X \xrightarrow{a}_{\mathcal{C}} X'} \chi_{X'}^{\mathrm{wh}} \right).$$

Here $\theta_X \in \mathrm{EIL}_{\mathrm{ro}}$ is as in Lemma 5.4. We further let $\chi_{\mathcal{C}}^{\mathrm{wh}} \stackrel{\mathrm{df}}{=} \chi_{\varnothing}^{\mathrm{wh}}$.

Note that $\chi_{\mathcal{C}}^{\mathrm{wh}} \in \mathrm{EIL}_{\mathrm{wh}}$ and $\mathsf{md}(\chi_X^{\mathrm{wh}}) \leqslant 2.c(\mathcal{C})$.

Proposition 6.14. Suppose Act is finite. Let $\mathcal{C}$ and $\mathcal{D}$ be finite stable configuration structures. Then $\mathcal{D} \approx_{\mathsf{wh}} \mathcal{C}$ if and only if $\mathcal{D} \models \chi_{\mathcal{C}}^{\mathrm{wh}}$.

Proof. The proof is similar to the proof of Proposition 6.12, except that it uses Theorem 5.15 instead of Theorem 5.12. □

Example 6.15. Recall from Example 3.26 that

$$a \,|\, a = (a \,|\, a) + a.a$$

holds for SB but not WH. Proposition 6.14 gives an alternative method for proving this: we define the WH characteristic formula χ for $a \,|\, a$ and argue that $(a \,|\, a) + a.a$ does not satisfy χ. The formula χ is defined in terms of subformulas χ_X, with one for each of the four configurations X of $a \,|\, a$, as in Definition 6.13. Since the two configurations consisting of a single event labelled a produce equivalent characteristic formulas, we have $\chi \equiv \chi_{\varnothing}$ where

$$\begin{aligned}
\chi_{\varnothing} &\equiv \langle a \rangle\rangle \chi_a \wedge \ [a]]\ \chi_a \\
\chi_a &\equiv (x : a)\langle\langle x \rangle \mathtt{tt} \ \wedge \ \langle a \rangle\rangle \chi_{a,a} \wedge \ [a]]\ \chi_{a,a} \\
\chi_{a,a} &\equiv (x : a)(y : a)(\langle\langle x \rangle \langle\langle y \rangle \mathtt{tt} \ \wedge \ \langle\langle x \rangle \mathtt{tt} \ \wedge \ \langle\langle y \rangle \mathtt{tt}).
\end{aligned}$$

The configuration structure $(a \,|\, a) + a.a$ does not satisfy χ because, unlike $a \,|\, a$, not all of its configurations with two events a satisfy $\chi_{a,a}$ (the two events a are independent).

7. Conclusions and future work

We have introduced a logic that uses event identifiers to track events in both forward and reverse directions. As we have seen, this enables it to express causality and concurrency between events. The logic is strong enough to characterise hereditary history-preserving (HH) bisimulation equivalence. We are also able to characterise the most well-known bisimulation-based weaker equivalences using sublogics. In particular, we can characterise weak history-preserving bisimulation, which has not been done previously as far as we are aware. We have also investigated characteristic formulas for our logic with respect to HH and other equivalences. Again, we are not aware of any previous work on characteristic formulas for logics for true concurrency.

In future work we would like to:

(1) investigate general laws that hold for the logic;

https://doi.org/10.1017/S0960129513000510 Published online by Cambridge University Press

(2) look at sublogics characterising other true concurrency equivalences, including equivalences involving reverse transitions from Bednarczyk (1991) and Phillips and Ulidowski (2012);
(3) consider the logic extended with recursion, and how more complex properties can be expressed using it; and
(4) answer the open question raised in Section 6 as to whether there is a single characteristic formula for a finite structure with respect to HH equivalence.

Acknowledgements

We are grateful to Ian Hodkinson and the anonymous referees of *Mathematical Structures in Computer Science* and of EXPRESS 2011 for helpful comments and suggestions.

References

Aceto, L., Ingólfsdóttir, A. and Sack, J. (2009) Characteristic formulae for fixed-point semantics: A general framework. In: Proceedings 16th International Workshop on Expressiveness in Concurrency, EXPRESS 2009. *Electronic Proceedings in Theoretical Computer Science* **8** 1–15.

Baldan, P. and Crafa, S. (2010) A logic for true concurrency. In: Proceedings of 21st International Conference on Concurrency Theory, CONCUR 2010. *Springer-Verlag Lecture Notes in Computer Science* **6269** 147–161.

Baldan, P. and Crafa, S. (2011) A logic for true concurrency. *CoRR*, abs/1110.4094.

Bednarczyk, M. A. (1991) Hereditary history preserving bisimulations or what is the power of the future perfect in program logics. Technical report, Institute of Computer Science, Polish Academy of Sciences, Gdańsk.

Boudol, G. and Castellani, I. (1987) On the semantics of concurrency: partial orders and transition systems. In: Proceedings of TAPSOFT '87. *Springer-Verlag Lecture Notes in Computer Science* **249** 123–137.

Cherief, F. (1992) Back and forth bisimulations on prime event structures. In: Proceedings of PARLE '92. *Springer-Verlag Lecture Notes in Computer Science* **605** 843–858.

Degano, P., De Nicola, R. and Montanari, U. (1987) Observational equivalences for concurrency models. In: Wirsing, M. (ed.) *Formal Descriptions of Programming Concepts – III, Proceedings of the 3rd IFIP WG 2.2 Conference*, North-Holland 105–129.

De Nicola, R. and Ferrari, G. L. (1990) Observational logics and concurrency models. In: Proceedings of 10th Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 1990. *Springer-Verlag Lecture Notes in Computer Science* **472** 301–315.

De Nicola, R., Montanari, U. and Vaandrager, F. (1990) Back and forth bisimulations. In: Proceedings of CONCUR '90, Theories of Concurrency: Unification and Extension. *Springer-Verlag Lecture Notes in Computer Science* **458** 152–165.

De Nicola, R. and Vaandrager, F. (1990) Three logics for branching bisimulation (extended abstract). In: *Proceedings, Fifth Annual IEEE Symposium on Logic in Computer Science, LICS 1990*, IEEE Computer Society Press 118–129.

Fecher, H. (2004) A completed hierarchy of true concurrent equivalences. *Information Processing Letters* **89** (5) 261–265.

Fröschle, S. B. (1999) Decidability of plain and hereditary history-preserving bisimilarity for BPP. In: Proceedings of 6th International Workshop on Expressiveness in Concurrency, EXPRESS '99. *Electronic Notes in Theoretical Computer Science* **27** 85–106.

Fröschle, S. B. (2005) Composition and decomposition in true-concurrency. In: Foundations of Software Science and Computational Structures, 8th International Conference, FOSSACS 2005. *Springer-Verlag Lecture Notes in Computer Science* **3441** 333–347.

Fröschle, S. B. and Lasota, S. (2005) Decomposition and complexity of hereditary history preserving bisimulation on BPP. In: Proceedings of 16th International Conference on Concurrency Theory, CONCUR 2005. *Springer-Verlag Lecture Notes in Computer Science* **3653** 263–277.

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

van Glabbeek, R. J. and Plotkin, G. D. (1995) Configuration structures. In: *Proceedings of 10th Annual IEEE Symposium on Logic in Computer Science, LICS 1995*, IEEE Computer Society Press 199–209.

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

van Glabbeek, R. J. and Vaandrager, F. W. (1997) The difference between splitting in n and $n+1$. *Information and Computation* **136** (2) 109–142.

Goltz, U., Kuiper, R. and Penczek, W. (1992) Propositional temporal logics and equivalences. In: Proceedings of 3rd International Conference on Concurrency Theory, CONCUR '92. *Springer-Verlag Lecture Notes in Computer Science* **630** 222–236.

Graf, S. and Sifakis, J. (1986) A modal characterization of observational congruence on finite terms of CCS. *Information and Control* **68** (1-3) 125–145.

Gutierrez, J. (2009) Logics and bisimulation games for concurrency, causality and conflict. In: Proceedings of the 12th International Conference on Foundations of Software Science and Computation Structures, FOSSACS 2009. *Springer-Verlag Lecture Notes in Computer Science* **5504** 48–62.

Gutierrez, J. and Bradfield, J. C. (2009) Model-checking games for fixpoint logics with partial order models. In: Proceedings of the 20th International Conference on Concurrency Theory, CONCUR 2009. *Springer-Verlag Lecture Notes in Computer Science* **5710** 354–368.

Hennessy, M. C. B. and Milner, R. (1985) Algebraic laws for nondeterminism and concurrency. *Journal of the Association for Computing Machinery* **32** (1) 137–161.

Hennessy, M. C. B. and Stirling, C. (1985) The power of the future perfect in program logics. *Information and Control* **67** 23–52.

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

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

Laroussinie, F., Pinchinat, S. and Schnoebelen, Ph. (1995) Translations between modal logics of reactive systems. *Theoretical Computer Science* **140** (1) 53–71.

Laroussinie, F. and Schnoebelen, Ph. (1995) A hierarchy of temporal logics with past. *Theoretical Computer Science* **148** 303–324.

Mukund, M. and Thiagarajan, P. S. (1992) A logical characterization of well branching event structures. *Theoretical Computer Science* **96** (1) 35–72.

Nielsen, M. and Clausen, C. (1994a) Bisimulation for models in concurrency. In: Proceedings of 5th International Conference on Concurrency Theory, CONCUR '94. *Springer-Verlag Lecture Notes in Computer Science* **836** 385–400.

Nielsen, M. and Clausen, C. (1994b) Bisimulation, games, and logic. In: Results and Trends in Theoretical Computer Science. *Springer-Verlag Lecture Notes in Computer Science* **812** 289–306.

Nielsen, M. and Clausen, C. (1995) Games and logics for a noninterleaving bisimulation. *Nordic Journal of Computing* **2** (2) 221–249.

Nielsen, M., Plotkin, G. D. and Winskel, G. (1981) Petri nets, event structures and domains, part I. *Theoretical Computer Science* **13** 85–108.

Penczek, W. (1995) Branching time and partial order in temporal logics. In: *Time and Logic: A Computational Approach*, UCL Press Ltd 179–228.

Phillips, I. C. C. and Ulidowski, I. (2006) Reversing algebraic process calculi. In: Proceedings of Ninth International Conference on Foundations of Software Science and Computation Structures, FOSSACS 2006. *Springer-Verlag Lecture Notes in Computer Science* **3921** 246–260.

Phillips, I. C. C. and Ulidowski, I. (2007) Reversing algebraic process calculi. *Journal of Logic and Algebraic Programming* **73** (1-2) 70–96.

Phillips, I. C. C. and Ulidowski, I. (2011) A logic with reverse modalities for history-preserving bisimulations. In: Proceedings of Eighteenth International Workshop on Expressiveness in Concurrency, EXPRESS 2011. *Electronic Proceedings in Theoretical Computer Science* **64** 104–118.

Phillips, I. C. C. and Ulidowski, I. (2012) A hierarchy of reverse bisimulations on stable configuration structures. *Mathematical Structures in Computer Science* **22** 333–372.

Pinchinat, S., Laroussinie, F. and Schnoebelen, Ph. (1994) Logical characterizations of truly concurrent bisimulation. Technical Report 114, Grenoble.

Pomello, L. (1986) Some equivalence notions for concurrent systems – An overview. In: Advances in Petri Nets 1985. *Springer-Verlag Lecture Notes in Computer Science* **222** 381–400.

Rabinovich, A. and Trakhtenbrot, B. A. (1988) Behavior structures and nets. *Fundamenta Informaticae* **11** (4) 357–403.

Steffen, B. and Ingólfsdóttir, A. (1994) Characteristic formulae for processes with divergence. *Information and Computation* **110** (1) 149–163.

Winskel, G. (1987) Event structures. In: Advances in Petri Nets 1986. *Springer-Verlag Lecture Notes in Computer Science* **255** 325–392.