Janicki R, Kleijn J, Koutny M, Mikulski L. Step Traces. *Acta Informatica* 2015. DOI: 10.1007/s00236-015-0244-z

Acta Informatica manuscript No.
(will be inserted by the editor)

Step traces

Ryszard Janicki · Jetty Kleijn · Maciej Koutny · Łukasz Mikulski

Received: date / Accepted: date

Abstract In the classical Mazurkiewicz trace approach the behaviour of a concurrent system is described in terms of sequential observations that differ only with respect to their ordering of independent actions. This paper investigates an extension of the trace model to the case that actions can be observed as occurring simultaneously. Thus observations are sequences of steps, i.e., sets of actions. This leads to a step trace model based on three relations between events: simultaneity, serialisability, and interleaving. Whereas the underlying causal structures of traces are based on dependencies between actions leading to a partial order interpretation, more general causal structures are needed to describe the invariant relationships between the action occurrences in a step trace. We present a complete picture including dependence structures extending dependence graphs, and a characterisation of step traces in terms of invariant order structures.

Keywords trace · step trace · causality · independence · simultaneity · serialisability · sequentialisability · interleaving · extending concurrency alphabets · dependence graph · dependence structure · invariant order structure

R.Janicki
Department of Computing and Software, McMaster University, Hamilton, ON, L8S 4K1, Canada, E-mail: janicki@mcmaster.ca

J.Kleijn
LIACS, Leiden University, P.O.Box 9512, NL-2300 RA Leiden, The Netherlands, E-mail: h.c.m.kleijn@liacs.leidenuniv.nll

M.Koutny
School of Computing Science, Newcastle University, Newcastle upon Tyne NE1 7RU, United Kingdom, E-mail: maciej.koutny@ncl.ac.uk

Ł.Mikulski
Fac. of Mathematics and Computer Science, Nicolaus Copernicus University, Toruń, Chopina 12/18, Poland, E-mail: lukasz.mikulski@mat.umk.pl

1 Introduction

Mazurkiewicz traces [28,30] are a well-established, classical, and basic model for representing and structuring sequential observations of concurrent behaviour; see, e.g., [6,23].

The fundamental assumption underlying trace theory is that independent events (occurrences of actions) may be observed in any order. Sequences that differ only w.r.t. their ordering of independent events are identified as belonging to the same concurrent run of the system under consideration. Thus a trace is an equivalence class of sequences comprising all (sequential) observations of a single concurrent run. The dependencies between the events of a trace are invariant among (common to) all elements of the trace. This (acyclic) dependence graph determines through its transitive closure the underlying causality structure of the trace as a (labelled) partial order [34]. In fact, this partial order can also be obtained as the intersection of the labelled total orders corresponding to the sequences forming the trace. Moreover, the linearisations (saturations) of this partial order correspond exactly to the sequences belonging to the trace. Thus a trace can be seen as a labelled partial order which is unique up to isomorphism, i.e., the names of the underlying elements; see, e.g., [6,9,23]. Paper [38] provides the necessary connection (Szpilrajn's property) between causal structures (partial orders) and observations (total orders), by showing that each partial order is the intersection of all its linearisations. The overall setup can be summarised by the schematic commuting diagram shown in Figure 1.

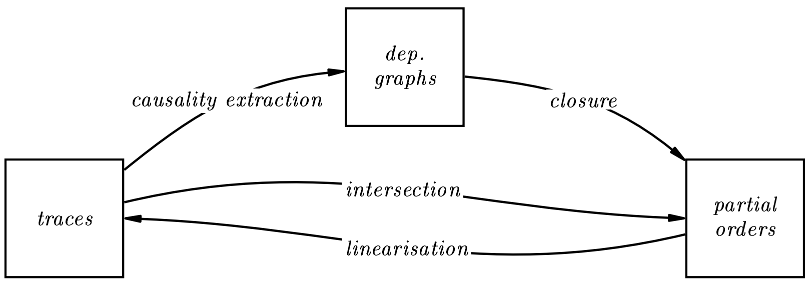


Fig. 1 Correspondence diagram for Mazurkiewicz traces.

Being based on equating independence and lack of ordering, the concurrency paradigm of Mazurkiewicz traces and the corresponding partial order interpretation of concurrency is rather restricted [16].

In this paper, we carefully consider how to extend the trace approach to a more general situation by assuming that observers may not only register the occurrence of one action before another, but can also record simultaneous occurrences of actions. Thus here observations consist of sequences of *steps*, i.e., sets of one or more actions that occur simultaneously. Still we aim at retaining the original philosophy underlying Mazurkiewicz traces and our setup will be based on just a few explicit and simple design choices. Our considerations lead to the concept of an extended concurrency alphabet with three basic relations between pairs of different actions:

simultaneity indicating that actions may occur together in a step; *serialisability* indicating a possible execution order for potentially simultaneous actions; and *interleaving* indicating that actions can *not* occur simultaneously though no specific ordering is required. These three relations can then be used to identify step sequences as observations of the same concurrent run. The resulting equivalence classes of step sequences are called *step traces*. It is the *main goal* of this paper to characterise such traces in terms of the underlying causal structures, in effect aiming to lift the diagram from Figure 1. Actually, as we will show, the generalisation that we obtain corresponds to the most general order structures, namely those associated with concurrent histories without any constraints in the sense of [16].
First however, we introduce a technically more convenient definition of step traces. It is based on the notion of a *step alphabet* with only two relations: simultaneity as before and *sequentialisability* which is a combination of serialisability and interleaving.
Next, we turn to the causal order structures underlying step traces with the ultimate aim to match step traces and step sequences with relational structures, just like Mazurkiewicz traces correspond to partial orders and total orders to sequences of action occurrences (see Figure 1). Partial orders are clearly not expressive enough to capture all possible relationships between events as determined by a step alphabet. Rather than a strict order (causality or 'before'), the relational structures we consider have a 'not later than' relation to represent weak causality (i.e., before or in the same step) and a 'mutual exclusion' relation for pure interleaving (not allowed in the same step but not causally ordered). Moreover, as shown in [16], weak causality and mutex are sufficient to represent the most general concurrent histories. We thus arrive at so-called *order structures*, labelled relational structures satisfying separability (akin to acyclicity) and label-orderedness (akin to lack of auto-concurrency) properties, as the counterpart of the dependence graphs underlying Mazurkiewicz traces. Step sequences correspond to saturated versions of these structures.

The order structures that satisfy a general variant of Szpilrajn's property (meaning that they can be obtained as the intersection of their saturated extensions) have been identified in [12] as general mutex order structures. Moreover, the closure of an order structure is a general mutex order structure. Thus we are left with the investigation of the properties of order structures obtained as dependence (order) structures from step sequences. As expected, equivalent step sequences define the same dependence structure. It is however less obvious that, conversely, any step sequence (saturated order structure) derived from the dependence structure of a step sequence is equivalent with that step sequence (belongs to the same history). Eventually, the problem is reduced to the case of 'thin' step sequences in which every step is minimal in the sense that it cannot be serialised into a sequence of smaller steps, because its actions have to occur simultaneously. Interestingly, this leads to a proof technique similar to the approach for Mazurkiewicz traces consisting of sequences.

The whole discussion culminates in the development of a commutative diagram shown in Figure 3 for the model of traces based on step sequence observations, which is a counterpart of that the schematic diagram of Figure 1 that captures the relationship between traces and causal partial orders.

1.1 Traces of step sequences

Mazurkiewicz traces stem from two elegant mathematical ideas which can be used to capture the essence of equivalence between different observations of the same run of a concurrent system. Both are based on a notion of independence between actions expressed as a binary relation ind. The first idea uses the concept of equations expressing partial commutativity of action occurrences as determined by the independence relation. As a result, sequences $wabu$ and $wbau$ of action occurrences are considered equivalent whenever $\langle a,b\rangle \in \mathsf{ind}$, irrespective of what w and u are. The second idea is the common partial order structure that underlies equivalent observations and is defined by the ordering of the occurrences of dependent actions. Thus, each trace, i.e., equivalence class of sequential observations, has a unique (up to isomorphism) labelled partial order as its signature.

Equations could, in general, be of the form $a_1 \ldots a_k = b_1 \ldots b_m$ where the a_i and b_j are actions with, e.g., $c = de$ as a particular example. However, the usefulness for concurrency theory of equations in this form is not obvious, unless there is an additional interpretation of the alphabet of actions which usually entails the need for operators. This, in particular, happens when, instead of sequences of actions, one considers sequences of sets of actions (or step sequences) together with the operation of set union.

The idea of considering equations on sets of actions generated by relations on actions has been used to define, e.g., comtraces [17,27], g-comtraces [18], and interval traces [19]. Comtraces are a special case of absorbing monoids in the terminology of [18] — i.e., they are quotient monoids over step sequences derived from equations of the form $AB = A \uplus B$ — with the equations being derived from two relations, sim and ser, respectively called simultaneity and serialisability. Likewise, g-comtraces are a special case of partially commutative absorbing monoids in the terminology of [18] — i.e., they are quotient monoids derived from equations of the form $AB = A \uplus B$ and $AB = BA$ — with the equations being derived from simultaneity and serialisability as well as interleaving, inl. As shown in [24], the equations used in [18] and the subsequent papers do not model the relevant aspects of concurrent behaviours in a fully adequate way. In essence, the problem was that the interleaving equations $AB = BA$ were defined only by $A \times B \subseteq \mathsf{inl}$, in effect disallowing the mixing of two different 'reasons' for commuting two actions; the other one being $A \times B \subseteq \mathsf{ser} \cap \mathsf{ser}^{-1}$ (for a detailed discussion see Section 3). The corresponding model of causal structures was also not completely satisfactory, and a suitable improvement was proposed in [12].

In this paper, we will take a fresh look at the way in which a theory of traces consisting of step sequences could be developed and, in particular, we will develop an improved treatment of equations on step sequences of [18]. The soundness of the proposed improvement will be demonstrated in the second part of the paper by showing how the recently proposed model of causal structures matches exactly the extension of Mazurkiewicz traces introduced here.

1.2 Contribution of the paper

The first contribution of this paper is a detailed discussion of what could be a basic extension of Mazurkiewicz' concurrency alphabets to the case of step sequence executions assuming that only swapping and splitting of steps can lead to equivalent executions. When it is further assumed that such operations on steps are based on binary relations over actions in systems which do not exhibit auto-concurrency, this leads to the introduction of step alphabets based on simultaneity and sequentialisability relations and step traces. We also introduce a partition of step alphabets into more detailed relations which capture fine details pertaining to the understanding and analysis of concurrency phenomena.

The second contribution is the development of a class of labelled relational structures, called order structures, and their subclasses which can be used to represent step sequences, causal dependencies between action occurrences in step sequences, and ultimately step traces. The latter solves an outstanding problem of finding an order-theoretic characterisation of traces of step sequences corresponding to the most general class of concurrent histories composed of step sequence observations in [16].

The third contribution is the introduction of an order-theoretic counterpart of step traces which simplifies proofs and streamlines the treatment of the correspondence between step traces and invariant order structures.

1.3 Outline of the paper

We start by making explicit some notions and notations used in this paper. Section 3 discusses the proposed extension of classical trace theory to a model supporting step sequence executions. In particular, it introduces the notions of step alphabet and step trace. In Section 4, we investigate order structures which capture the causality in step sequences and step traces in the form of saturated order structures and invariant order structures, respectively. Section 5 brings together the extended model of traces and the extended model of causal order structures. The nature of the resulting correspondence is similar to that conveyed by the schematic diagram in Figure 1. Moreover, the notion of an order structure trace (a counterpart of step trace in the domain of order structures) is introduced. Section 6 concludes the paper, compares the approach developed here with other existing extensions of trace theory, and finally sketches possible directions for future work. The proofs of the formal results are included in the appendix.

2 Preliminaries

We use standard notions of set and formal language theory. Throughout the paper,

$$\boxed{\Sigma \text{ is an } \textit{action alphabet}} \quad \text{and} \quad \boxed{\mathbb{S} \text{ is the set of } \textit{steps} \text{ over } \Sigma}$$

We assume that Σ is finite and nonempty, and $\mathbb{S}$ comprises all nonempty sets of actions from Σ. SEQ will denote all finite sequences of actions (*sequences* over Σ), and SSEQ all finite sequences of steps (*step sequences* over Σ). We identify a singleton step with its only member, and non-singleton steps will be denoted by listing their elements within parentheses. Thus a step sequence $\{a\}\{b,c\}\{a\}$ can be written down as $a(bc)a$ or $a(cb)a$.

Let $u = A_1 \ldots A_k \in \mathbb{S}^*$ be a step sequence. Then:

- $\#_u(a)$ is the number of occurrences of an action a within u;
- $occ(u) = \{\langle a, i\rangle \mid a \in \Sigma \wedge 1 \le i \le \#_u(a)\}$ are the *action occurrences* of u;
- the *position* $pos_u(\alpha)$ within u of an action occurrence $\alpha = \langle a, i\rangle \in occ(u)$ is the smallest index $j \le k$ such that $\#_{A_1 \dots A_j}(a) = i$;
- $occ_i(u) = \{\alpha \in occ(u) \mid pos_u(\alpha) = i\}$ are the action occurrences contributing to the i-th step of u; and
- $occseq(u) = occ_1(u) \dots occ_k(u)$ is u with explicit action occurrences.

For example, $occ(a(bc)a) = \{\langle a, 1\rangle, \langle a, 2\rangle, \langle b, 1\rangle, \langle c, 1\rangle\}$, $pos_{a(bc)a}(\langle a, 2\rangle) = 3$, and $occseq(a(bc)a) = \{\langle a, 1\rangle\}\{\langle b, 1\rangle, \langle c, 1\rangle\}\{\langle a, 2\rangle\}$.

Let EQ be a finite set of equations on step sequences, each equation being of the form $u = v$, where u and v are nonempty step sequences. EQ induces a relation $\approx_{EQ}$ on step sequences comprising all pairs $\langle tuw, tvw\rangle$ such that $t, w \in \mathbb{S}^*$, and $u = v$ or $v = u$ is an equation in EQ. Furthermore, $\equiv_{EQ}$ is the equivalence relation on step sequences defined as $\approx_{EQ}^*$.

$X \xrightarrow{f} Y$ denotes a mapping f from X to Y, and $X' \xrightarrow{f} Y'$ the restriction of f to the domain $X' \subseteq X$ and codomain $Y' \supseteq f^{-1}(X')$.

For a binary relation R over X, $R^{sym} = R \cup R^{-1}$ denotes the symmetric closure, $R^{\curlywedge} = R^* \setminus id_X$ the irreflexive transitive closure, and $R^{\circledast} = R^* \cap (R^*)^{-1}$ the largest equivalence relation contained in R^*. R is a partial order relation if it is irreflexive and transitive, and a total order relation if it is a partial order relation such that $R^{sym} = (X \times X) \setminus id_X$.

A *labelled partial order* is a triple $po = \langle \Delta, \prec, \ell\rangle$, where $\Delta \xrightarrow{\ell} \Sigma$ is a labelling of the finite domain Δ and $\prec$ is a partial order relation on Δ. po is *total* if $\prec$ is a total partial order relation, and *label-linear* if $x \prec^{sym} y$, for all distinct $x, y \in \Delta$ satisfying $\ell(x) = \ell(y)$ (in such case all the elements with the same label are totally ordered by $\prec$).

3 Extending Mazurkiewicz traces

One of the aims of Mazurkiewicz trace theory is to add structure to the otherwise plain set of observations of the behaviour of a concurrent system, each observation being represented by a sequence of action occurrences. Action occurrences are assumed to be atomic and, crucially, there is a (static) notion of independence between pairs of actions. This independence relation is then used to identify observations which differ only by the order of occurrences of independent actions. The resulting equivalence relation groups together observations of the same concurrent run (history), with the corresponding equivalences classes being called *traces*. The relevance of the resulting framework is reinforced by the fact that it corresponds to an order theoretic model of partial order histories of concurrent systems and concurrent system models.

Here we aim at capturing possibly lightest extension of the theory of Mazurkiewicz traces in the case that the smallest unit of observation is a set of action occurrences (a step) rather than a single action occurrence, reflecting the idea that actions could occur (and be observed as occurring) simultaneously. Thus behaviour observations are now represented by step sequences rather than sequences of action occurrences. We will now elaborate our proposed extension, retaining the philosophy behind the original model, making explicit all key design choices, and motivating all specific design decisions.

The first design decision we face is what should be the form of the equations used in the extended model. *Interleaving* equations

$$\boxed{AB = BA \quad \text{with} \quad A \cap B = \varnothing}$$

directly generalise Mazurkiewicz's $ab = ba$, and so we will use them in the extended framework. But restricting ourselves to only interleaving equations would effectively turn the resulting traces of step sequences into a class of Mazurkiewicz traces with actions being sets. Then, for example, we would not be able to derive $(ab) = ab$ for two completely independent actions, a and b. We will therefore use in the extended framework *serialisation* equations

$$\boxed{C = DE \quad \text{with} \quad D \cap E = \varnothing}$$

allowing one to split a step into two consecutive substeps. *No other equations will be used nor needed.*

Note 1 We assumed $A \cap B = \varnothing$ as the order of different occurrences of an action should not be changed, and $D \cap E = \varnothing$ as in equivalent observations each action should occur the same number of times. ◇

We are only interested in those combinations of interleaving and serialisation equations which follow from the fundamental principle of Mazurkiewicz's approach which is that *all equivalences between behaviours are ultimately derived from binary relationships between actions.* Hence one needs to provide relation(s) on actions which determine what steps can be interleaved, what steps can be split and how, and indeed what actions can form legal steps. Our way to meet these requirements is to introduce three irreflexive binary relations over Σ:

- sim is a symmetric *simultaneity* relation defining all legal steps $A \in \mathbb{S}$ through $(A \times A) \setminus id_\Sigma \subseteq \mathsf{sim}$.
- inl is a symmetric relation defining the interleaving equations $AB = BA$ through $A \times B \subseteq \mathsf{inl}$. We will also require $\mathsf{inl} \cap \mathsf{sim} = \varnothing$, i.e., at this point interleaved action occurrences cannot be simultaneous (but see the discussion below).
- ser is a sub-relation of sim defining the serialisability equations $C = DE$ through $D \times E \subseteq \mathsf{ser}$ and $C = D \cup E$.

Suppose now that $A \times B \subseteq \mathsf{ser}$ and $B \times A \subseteq \mathsf{ser}$. Then, according to the above, we obtain two equations, $A \cup B = AB$ and $B \cup A = BA$. These equations, in turn, can be used to derive a new equivalence $AB = BA$. Hence, intuitively, for all pairs of actions $\langle a, b\rangle \in A \times B$ it is possible to commute. Taking this observation further, we will stipulate that $AB = BA$ provided that for all pairs of actions $\langle a, b\rangle \in A \times B$ it is the case that $\langle a, b\rangle \in \mathsf{inl}$ or $\langle a, b\rangle \in \mathsf{ser} \cap \mathsf{ser}^{-1}$. The interleaving equations $AB = BA$ will therefore be defined through $A \times B \subseteq \mathsf{inl} \cup (\mathsf{ser} \cap \mathsf{ser}^{-1})$ rather than by $A \times B \subseteq \mathsf{inl}$. This concludes the design of our extended trace model, with the three relations described above being the basic building blocks of the extended concurrency alphabets.

A *sim-inl-ser alphabet* is a quadruple $\psi = \langle \Sigma, \mathsf{sim}, \mathsf{inl}, \mathsf{ser}\rangle$, where $\mathsf{sim}, \mathsf{inl}, \mathsf{ser}$ are irreflexive relations over Σ such that sim and inl are symmetric, $\mathsf{inl} \cap \mathsf{sim} = \varnothing$, and $\mathsf{ser} \subseteq \mathsf{sim}$. All sim-inl-ser alphabets are denoted by Ψ.

The set $\mathbb{S}_\psi$ of *steps* defined by a sim-inl-ser alphabet ψ comprises all nonempty $A \subseteq \Sigma$ such that $\langle a, b\rangle \in \mathsf{sim}$, for all distinct $a, b \in A$. Moreover, the *equations* EQ_ψ induced by ψ are as follows, where $A, B \in \mathbb{S}_\psi$:

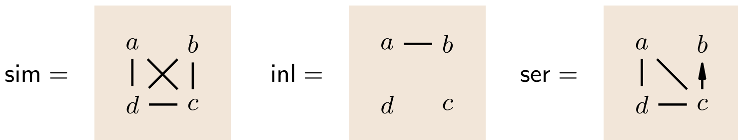
$$\begin{array}{lll} AB =_\psi BA & \text{if } A \times B \subseteq \mathsf{inl} \cup (\mathsf{ser} \cap \mathsf{ser}^{-1}) & (\textit{interleaving}) \\ AB =_\psi A \cup B & \text{if } A \times B \subseteq \mathsf{ser} & (\textit{serialisability}) \end{array}$$

The resulting relations $\approx_{EQ_\psi}$ and $\equiv_{EQ_\psi}$ on step sequences will respectively be denoted by $\approx_\psi$ and $\equiv_\psi$. The set of equivalence classes of $\equiv_\psi$ which contain at least one step sequence in $\mathsf{SSEQ}_\psi = \mathbb{S}_\psi^*$, called *sim-inl-ser traces*, will be denoted by STR_ψ. Moreover, the sim-inl-ser trace containing $u \in \mathsf{SSEQ}_\psi$ will be denoted by $[\![u]\!]_\psi$.

Applying the equations in EQ_ψ to step sequences composed of legal steps can never produce an illegal step.

Proposition 1 *If* $\tau \in \mathsf{STR}_\psi$ *then* $\tau \subseteq \mathsf{SSEQ}_\psi$.

Example 1 Consider $\psi_0 = \langle \{a, b, c, d\}, \mathsf{sim}, \mathsf{inl}, \mathsf{ser}\rangle$, a sim-inl-ser alphabet with simultaneity, interleaving, and serialisability relations given below, where each edge stands for two arrows in opposite directions:



ψ_0 generates, e.g., the interleaving equations $ab =_{\psi_0} ba$ and $a(bd) =_{\psi_0} (bd)a$, and serialisability equations $(ac) =_{\psi_0} ac$, $(ac) =_{\psi_0} ca$, and $(bc) =_{\psi_0} cb$. We also have:

$$\begin{array}{ll} [\![(bd)c]\!]_{\psi_0} & = \{(bd)c\} \\ [\![c(bd)]\!]_{\psi_0} & = \{c(bd), (cbd)\} \\ [\![a(bd)]\!]_{\psi_0} & = \{a(bd), (bd)a\} \\ [\![a(bc)]\!]_{\psi_0} & = \{a(bc), (bc)a, acb, cba, cab, (ca)b\} \\ [\![a(cd)]\!]_{\psi_0} & = \{(acd), a(cd), (ac)d, (ad)c, c(ad), (cd)a, d(ac), acd, adc, cad, cda, \\ & \qquad dac, dca\} \\ [\![a(bcd)]\!]_{\psi_0} & = \{a(bcd), (bcd)a, ac(bd), ca(bd), c(bd)a, (ac)(bd)\} \end{array}$$

We also note that $(bc) =_{\psi_0} bc$ is not an equation generated by ψ_0. $\diamond$

The above direct capture of an extended concurrency alphabet can be replaced by a simpler notion, based on two rather than three relations on actions: one is sim as above defining all legal steps, whereas the other one, seq, combines serialisability and (pure) interleaving.

Definition 1 (step alphabet) A *step alphabet* is a triple $\theta = \langle \Sigma, \mathsf{sim}, \mathsf{seq}\rangle$, where sim and seq, respectively called *simultaneity* and *sequentialisability*, are irreflexive relations over Σ. Moreover, sim and $\mathsf{seq} \setminus \mathsf{sim}$ are assumed to be symmetric. All step alphabets are denoted by Θ. $\diamond$

The set $\mathbb{S}_\theta$ of *steps* defined by a step alphabet θ comprises all nonempty $A \subseteq \Sigma$ such that $\langle a, b\rangle \in \mathsf{sim}$, for all distinct $a, b \in A$. Moreover, the *equations* EQ_θ induced by θ are as follows, where $A, B \in \mathbb{S}_\theta$:

$$\begin{array}{|llr|} AB =_\theta BA & \text{if } A \times B \subseteq \mathsf{seq} \cap \mathsf{seq}^{-1} & (\textit{interleaving}) \\ AB =_\theta A \cup B & \text{if } A \times B \subseteq \mathsf{seq} \cap \mathsf{sim} & (\textit{sequentialising}) \end{array}$$

The resulting relations $\approx_{EQ_\theta}$ and $\equiv_{EQ_\theta}$ on step sequences will respectively be denoted by $\approx_\theta$ and $\equiv_\theta$.

Definition 2 (step trace) A *step trace* over a step alphabet θ is an equivalence classes of $\equiv_\theta$ containing at least one step sequence in $\mathsf{SSEQ}_\theta = \mathbb{S}_\theta^*$. All such step traces over θ are denoted by STR_θ. Moreover, the step trace containing $u \in \mathsf{SSEQ}_\theta$ will be denoted by $[\![u]\!]_\theta$. $\diamond$

Applying the equations in EQ_θ to step sequences composed of legal steps can never produce an illegal step.

Proposition 2 *If* $\tau \in \mathsf{STR}_\theta$ *then* $\tau \subseteq \mathsf{SSEQ}_\theta$.

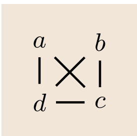
The two representations of extended concurrency alphabets, viz. sim-inl-ser alphabets and step alphabets, are equivalent in the sense that the traces defined are the same. We show this using the following two mappings:

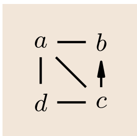
$$\begin{array}{|ll|} \Psi \xrightarrow{\mathsf{sa2gca}} \Theta & \langle \Sigma, \mathsf{sim}, \mathsf{inl}, \mathsf{ser}\rangle \mapsto \langle \Sigma, \mathsf{sim}, \mathsf{inl} \cup \mathsf{ser}\rangle \\ \Theta \xrightarrow{\mathsf{gca2sa}} \Psi & \langle \Sigma, \mathsf{sim}, \mathsf{seq}\rangle \mapsto \langle \Sigma, \mathsf{sim}, (\mathsf{seq} \cap \mathsf{seq}^{-1}) \setminus \mathsf{sim}, \mathsf{seq} \cap \mathsf{sim}\rangle \end{array}$$

Theorem 1 $\Psi \xrightarrow{\mathsf{sa2gca}} \Theta \xrightarrow{\mathsf{gca2sa}} \Psi$ *are inverse bijections such that, for all* $\psi \in \Psi$ *and* $\theta \in \Theta$, $\mathsf{STR}_{\mathsf{sa2gca}(\psi)} = \mathsf{STR}_\psi$ *and* $\mathsf{STR}_{\mathsf{gca2sa}(\theta)} = \mathsf{STR}_\theta$ *(i.e., the two mappings are trace-preserving).*

From this point on, for ease of reference, we may refer to the traces of step sequences defined by sim-inl-ser alphabets and step alphabet as *step traces*.

Example 2 The step alphabet corresponding to the sim-inl-ser alphabet ψ_0 of Example 1 has the following simultaneity and sequentialising relations:

$\mathsf{sim} =$ 

$\mathsf{seq} =$ 

In the last part of this section, we take another look at the structure of a step alphabet $\psi = \langle \Sigma, \mathsf{sim}, \mathsf{seq}\rangle \in \Psi$. We then single out six semantically meaningful relationships between pairs of actions which form a *partition* of $\Sigma \times \Sigma$ (see [31] for a similar partition in the case of comtraces):

- ssi defined as $\mathsf{sim} \setminus (\mathsf{seq} \cup \mathsf{seq}^{-1})$ is *strong simultaneity* allowing a pair of actions to be executed simultaneously, and disallowing sequentialisability and interleaving.

sse defined as $(\mathsf{seq} \setminus \mathsf{seq}^{-1}) \cap \mathsf{sim}$ is *semi-sequentialisability* allowing a pair of simultaneously executed actions to be executed in the order given, but not in the reverse order.

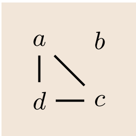
con defined as $\mathsf{seq} \cap \mathsf{seq}^{-1} \cap \mathsf{sim}$ is *concurrency* identifying actions which can be executed simultaneously as well as in any order.

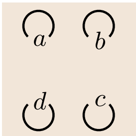
wdp defined as $(\mathsf{seq}^{-1} \setminus \mathsf{seq}) \cap \mathsf{sim}$ is *weak dependence* which is an inverse of semi-sequentialisability.

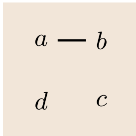
rig defined as $(\Sigma \times \Sigma) \setminus (\mathsf{sim} \cup (\mathsf{seq} \cap \mathsf{seq}^{-1}))$ is *rigid order* allowing neither simultaneity nor changing of the order of actions.

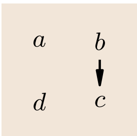
inl defined as $(\mathsf{seq} \cap \mathsf{seq}^{-1}) \setminus \mathsf{sim}$ is *interleaving* as before.

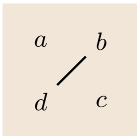
Example 3 For the step alphabet of Example 1, the relations derived above are as follows:

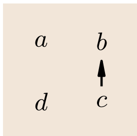
con = 

rig = 

inl = 

wdp = 

ssi = 

sse = 

Hence a and b are the only truly interleaved actions, while b and d are the only actions whose sequentialisation and interleaving is disallowed (this does not prevent b and d from occurring in the same step). The rigid order, which plays the role of dependence in Mazurkiewicz trace theory, is implied by label-linearity and does not involve any pair of different actions. ◇

4 Extending causal structures

This section describes *order structures*, a class of labelled relational structures which will be used to represent the observational and causal relationships in the behaviours of concurrent systems. Also introduced are *saturated* order structures (SO-structures) that represent individual step sequence observations, and *invariant* order structures (IO-structures) that represent causal relationships underpinning step traces. The main goal is to identify relational structures matching step traces and step sequences in the same way as partial orders match Mazurkiewicz traces, and total orders match sequences of action occurrences.

4.1 Order structures

We start the order theoretic treatment of step traces by formally introducing *relational structures* and formulating at this level properties which are essential for the definition of order structures. The first property is separability which corresponds to acyclicity in the domain of binary relations, and the second label-orderedness which corresponds to label-linearity introduced in Section 2 for labelled partial orders. The latter property will turn out to be a powerful notion which essentially allows one to completely abstract from the identities of the underlying domain elements.

A *relational structure* is a triple $rs = \langle \Delta, \rightleftharpoons, \sqsubset, \ell \rangle$, where $\Delta \xrightarrow{\ell} \Sigma$ is a labelling of a finite domain Δ, and $\rightleftharpoons, \sqsubset$ are two binary relations on Δ. We interpret Δ as the set of events that have happened, $x \rightleftharpoons y$ as a record that x occurred *not simultaneously* with y, and $x \sqsubset y$ that x occurred *not later* than y, i.e., *before or simultaneously* with y. The relations $\rightleftharpoons$ and $\sqsubset$ will therefore be respectively called *mutex* and *weak causality*. Moreover, if both $x \sqsubset y$ and $x \rightleftharpoons y$ hold, then x must have occurred *before* y. For this reason, we will refer to the intersection of $\sqsubset$ and $\rightleftharpoons$ as *causality* (or *precedence*), denoting it by $\prec$. The labelling function ℓ associates an action with each event, with distinct events corresponding to distinct occurrences (or executions) of actions. For every label a, we will use $\prec^a$ to denote $\prec$ restricted to the elements labelled with a, and write Δ_{rs} $\rightleftharpoons_{rs}$, etc, to emphasize the relational structure rs.

The properties relevant to the relations between action occurrences are defined as follows. A relational structure $rs = \langle \Delta, \rightleftharpoons, \sqsubset, \ell \rangle$ is:

- *separable* if $\rightleftharpoons$ is symmetric, $\sqsubset$ is irreflexive, and $\rightleftharpoons \cap \sqsubset^{\circledast} = \varnothing$ (note that this implies that $\rightleftharpoons$ is also irreflexive as id_Δ is included in $\sqsubset^{\circledast}$);
- *label-ordered* if $x \prec y$ or $y \prec x$, for all $x \neq y$ satisfying $\ell(x) = \ell(y)$; and
- *label-linear* if $\prec^a$ is a total order relation, for every label $a \in \Sigma$.

Label-orderedness guarantees that domain elements with the same label (intuitively representing two occurrences of the same action) are related by $\prec$.

Proposition 3 *Every separable label-ordered relational structure is label-linear.*

We can now introduce a notion which is central to our treatment of step traces.

Definition 3 (order structure) An *order structure* is a separable and label-ordered relational structure. All order structures are denoted by OS. ◇

Since $x \sqsubset y \sqsubset x$ means that x and y are *simultaneous* events, the requirement of separability excludes situations where events forming a weak causality cycle — captured by $\sqsubset^{\circledast}$ — are also involved in the mutex relationship. Label-orderedness together with separability guarantees that all events labelled by the same action are totally ordered, hence order structures are *label-linear* (see Proposition 3).

Note 2 Referring to the setup of Mazurkiewicz traces, order structures correspond to acyclic relations.

4.2 Isomorphism, extension, and closure

We will now introduce and discuss some properties of relational structures which are direct counterparts of similar notions in the domain of binary relations and partial orders.

Let $rs = \langle \Delta, \rightleftharpoons, \sqsubset, \ell \rangle$ and $rs' = \langle \Delta', \rightleftharpoons', \sqsubset', \ell' \rangle$ be two relational structures.

- rs and rs' are *isomorphic* if there exists a label-preserving bijection $\Delta \xrightarrow{\kappa} \Delta'$ such that $x \rightleftharpoons y$ iff $\kappa(x) \rightleftharpoons' \kappa(y)$, and $x \sqsubset y$ iff $\kappa(x) \sqsubset' \kappa(y)$, for all $x, y \in \Delta$. We denote this by $rs \sim_\kappa rs'$ or $rs \sim rs'$.
- rs' is an *extension* of rs if $\Delta = \Delta'$, $\ell = \ell'$, $\rightleftharpoons \subseteq \rightleftharpoons'$ and $\sqsubset \subseteq \sqsubset'$. We denote this by $rs' \in \mathsf{ext}(rs)$ or $rs \lhd rs'$.

Isomorphisms between label-linear relational structures are unique.

Proposition 4 *If there is a bijection establishing an isomorphism between two label-linear relational structures, then it is unique.*

The next notion we introduce is *structure-closure* which is a counterpart of the transitive closure of an acyclic relation. It is defined in terms of two families of relational structures, one being a subset of the other. Structures belonging to the smaller family are closed and the closure of a structure belonging to the larger family is obtained by extending its component relations leading to a structure in the smaller family.

Given two families of relational structures, $\mathsf{F} \supset \mathsf{F}'$, a *structure-closure operator of* F *with respect to* F' is a mapping $\mathsf{F} \xrightarrow{\mathsf{cls}} \mathsf{F}'$ such that, for all $rs \in \mathsf{F}$ and $rs' \in \mathsf{F}'$:

$$rs \lhd \mathsf{cls}(rs) \tag{1}$$

$$rs \lhd rs' \implies \mathsf{cls}(rs) \lhd rs' \; . \tag{2}$$

We then obtain that closing a closed structure has no effect, and all the closed extensions of a relational structure are also extensions of the closure of that structure, i.e., closing a structure does not enlarge 'too much' the component relations.

Proposition 5 *Let* $\mathsf{F} \xrightarrow{\mathsf{cls}} \mathsf{F}'$ *be a structure-closure operator. Then, for all* $rs \in \mathsf{F}$ *and* $rs' \in \mathsf{F}'$*:*

1. $\mathsf{cls}(rs') = rs'$.
2. $\mathsf{ext}(rs) \cap \mathsf{F}' = \mathsf{ext}(\mathsf{cls}(rs)) \cap \mathsf{F}'$.

The final notion we introduce in this section will be used to provide an order-theoretic counterpart of a set of step sequences belonging to some step trace.

A nonempty set rss of relational structures is *consistent* if all these relational structures have the same domain Δ and domain labelling ℓ. For such a set, the *intersection* is the relational structure:

$$\bigcap rss = \langle \Delta, \bigcap_{rs \in rss} \rightleftharpoons_{rs}, \bigcap_{rs \in rss} \sqsubset_{rs}, \ell \rangle \; .$$

A consistent rss is said to be *separable* or *label-ordered* or *label-linear* if so is $\bigcap rss$. In this paper, we will be interested in consistent sets of label-linear relational structures.

Directly from the definitions, we obtain:

Proposition 6 *Let rss be a consistent set of relational structures.*

1. *If rss is label-ordered, then so are all its elements.*
2. *If at least one element of rss is separable, then so is rss.*

The implications in the above proposition cannot be reversed. Moreover, it is not the case that the relational structures belonging to a label-linear *rss* have to be label-linear.

Proposition 7 *Let rss be a label-linear consistent set of label-linear relational structures, and $a \in \Sigma$ be a label. Then $\prec^a_{\bigcap rss} = \prec^a_{rs}$, for all $rs \in rss$.*

Two label-linear consistent sets of label-linear relational structures, *rss* and *rss'*, are *isomorphic* if there are bijections $\Delta_{rss} \xrightarrow{\kappa} \Delta_{rss'}$ and $rss \xrightarrow{\phi} rss'$ such that $rs \sim_\kappa \phi(rs)$, for all $rs \in rss$. We denote this by $rss \sim rss'$ or $rss \sim_{\kappa,\phi} rss'$.

Proposition 8 *Two label-linear consistent sets rss and rss' of label-linear relational structures are isomorphic if and only if for each relational structure in one set there is an isomorphic relational structure in the other set.*

Using similar arguments as in the proof of Proposition 8, we obtain the uniqueness of isomorphisms between label-linear consistent sets of label-linear relational structures.

Proposition 9 *If there are bijections (κ, ϕ) establishing an isomorphism between two label-linear consistent sets rss and rss' of label-linear relational structures, then each of them is unique.*

4.3 Saturated order structures

An order structure representing a single step sequence observation has to have all the observational relationships between events determined, i.e., it needs to be $\lhd$-maximal within the set of order structures.

Definition 4 (saturated order structure) An order structure *os* is *saturated* if $\mathsf{ext}(os) \cap \mathsf{OS} = \{os\}$. All saturated order structures (SO-structures) are denoted by SOS. $\diamond$

In the original definition of saturated order structures in [12], label-orderedness was not an issue as only unlabelled structures were considered there.

Note 3 Referring to the setup of Mazurkiewicz traces, saturated order structures correspond to total orders where adding an additional ordering between two elements destroys acyclicity. In the case of a saturated order structure, adding extra mutex or weak causality relations between events destroys separability.

Knowing only that an order structure is saturated is not very useful when it comes to proofs and understanding of other properties. Therefore, we will now provide an axiomatic characterisation of saturated order structures.

Proposition 10 *A relational structure $\langle \Delta, \rightleftharpoons, \sqsubset, \ell \rangle$ is saturated if and only if*

$$
\begin{array}{rrcll}
x \neq y \ \wedge & x \sqsubset z \sqsubset y & \Longrightarrow & x \sqsubset y & : \ L1 \\
 & x \rightleftharpoons y & \Longrightarrow & x \sqsubset^{sym} y & : \ L2 \\
x \neq y \ \wedge & x \not\rightleftharpoons y & \Longleftrightarrow & x \sqsubset y \sqsubset x & : \ L3 \\
x \neq y \ \wedge & \ell(x) = \ell(y) & \Longrightarrow & x \rightleftharpoons y & : \ L4
\end{array}
$$

for all $x, y, z \in \Delta$.

Intuitively, e.g., Axiom *L2* means that if events x and y are not simultaneous, then one of them must have happened before the other. Moreover, Axioms *L2* and *L4* together imply label-orderedness.

4.4 Invariants and histories

Within the order-theoretic part of Mazurkiewicz' approach, there are two ways in which one can represent concurrent behaviour: by means of a causal partial order *po* (or a causal *invariant* in the terminology of [16]); and through a set of total orders T which are the sequential observations of *po* (or a *history* in the terminology of [16]). These two representations are in one-to-one correspondence; more precisely, T is obtained by linearising *po* in all possible ways, and *po* can be obtained from T by intersecting the total orders it contains. This combination of invariant/history has been adopted in [16], where a general notion of history and underlying invariants were proposed. We will now revisit the resulting framework for the model of order structures.

Following the general approach, we consider two ways of representing a history. An order structure (a dependence structure), typically non-saturated, captures the causal invariants underlying the history, whereas a set of saturated order structures captures the observations of the history. Of course, not any combination of SO-structures represents a concurrent history. Below we assume that all SO-structures involved have at least *the same action occurrences* and *the same ordering* of the occurrences of any given action.

Definition 5 (SO-structure set) An SO-*structure set* is a label-ordered consistent set of SO-structures. All SO-structure set (SOS-sets) are denoted by SOSS. $\diamond$

In other words, the SO-structures belonging to an SOS-set share their domain and, in addition, induce the same total ordering on events labelled by any given action (see Propositions 3, 6 and 7).

To move between SOS-sets (histories) and order structures (invariants) we use the operations of *intersection*, $\mathsf{soss2os}$, and *saturation*, $\mathsf{os2soss}$:

$$\begin{array}{llll} \mathsf{SOSS} \xrightarrow{\mathsf{soss2os}} \mathsf{OS} & & soss \mapsto \bigcap soss \\ \mathsf{OS} \xrightarrow{\mathsf{os2soss}} \mathsf{SOSS} & & os \mapsto \mathsf{ext}(os) \cap \mathsf{SOS} \end{array} \tag{3}$$

Proposition 11 *The mappings* $\mathsf{soss2os}$ *and* $\mathsf{os2soss}$ *are well-defined.*

We are now in a position to state what it means that an order structure is an invariant, and that an SOS-set is a history.

Definition 6 (invariant) An *invariant order structure* is $ios \in \mathsf{OS}$ satisfying $ios = \mathsf{soss2os} \circ \mathsf{os2soss}(ios)$.[1] All invariant order structure (IO-structures) are denoted by IOS. $\diamond$

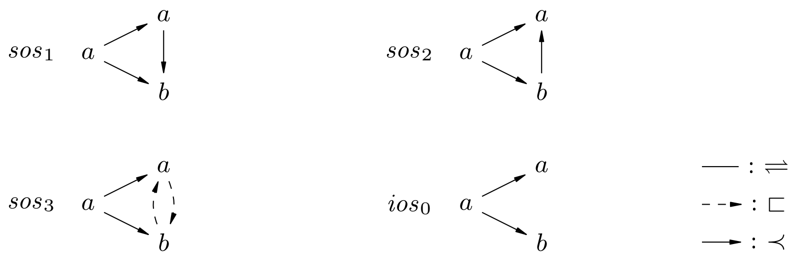
[1] Note that $\mathsf{os2soss}(ios) \neq \varnothing$ holds by Proposition 11.

The equality $ios = \mathsf{soss2os} \circ \mathsf{os2soss}(ios)$ is a version of Szpilrajn's property [38], which states that a poset is the intersection of its total order extensions, and plays a key role in the model of Mazurkiewicz traces.

Note 4 Referring to the setup of Mazurkiewicz traces, invariant order structures correspond to labelled partial orders.

Definition 7 (history) A *history* SOS-*set* is $hsoss \in \mathsf{SOSS}$ satisfying $hsoss = \mathsf{os2soss} \circ \mathsf{soss2os}(hsoss)$. All history SOS-sets (HSOS-sets) are denoted by HSOSS. $\diamond$

Example 4 Consider three SO-structures, sos_i $(i = 1, 2, 3)$, and an order structure, ios_0, depicted below:



Then $hsoss_0 = \{sos_1, sos_2, sos_3\}$ is a history SOS-set and ios_0 is an invariant order structure such that $ios_0 = \mathsf{soss2os}(hsoss_0)$ and $hsoss_0 = \mathsf{os2soss}(ios_0)$. $\diamond$

IO-structures are the causal invariants in the realm of order structures and, according to the next result, their sets of saturated extensions are concurrent histories.

Theorem 2 $\mathsf{IOS} \xrightarrow{\mathsf{os2soss}} \mathsf{HSOSS} \xrightarrow{\mathsf{soss2os}} \mathsf{IOS}$ *are inverse bijections.*

An axiomatic characterisation of invariant order structures without domain labellings was introduced in [12]. In the next definition we augment this characterisation with Axiom *I7* to ensure label-linearity, obtaining a complete axiomatisation of invariant order structures.

Theorem 3 *A relational structure* $\langle \Delta, \rightleftharpoons, \sqsubset, \ell \rangle$ *is an invariant order structure if and only if*

$$\begin{array}{rcll}
 & & x \not\sqsubset x & : \; I1 \\
x \neq y \;\wedge\; x \sqsubset z \sqsubset y & \Longrightarrow & x \sqsubset y & : \; I2 \\
x \rightleftharpoons y & \Longrightarrow & y \rightleftharpoons x \neq y & : \; I3 \\
x \prec z \sqsubset y \;\vee\; x \sqsubset z \prec y & \Longrightarrow & x \rightleftharpoons y & : \; I4 \\
z \rightleftharpoons y \;\wedge\; z \sqsubset x \sqsubset z & \Longrightarrow & x \rightleftharpoons y & : \; I5 \\
z \rightleftharpoons z' \;\wedge\; x \sqsubset z \sqsubset y \;\wedge\; x \sqsubset z' \sqsubset y & \Longrightarrow & x \rightleftharpoons y & : \; I6 \\
x \neq y \;\wedge\; \ell(x) = \ell(y) & \Longrightarrow & x \prec^{sym} y & : \; I7
\end{array}$$

for all $x, y, z, z' \in \Delta$

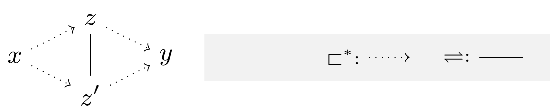
Order structures are like dependence graphs (acyclic relations) in the model of Mazurkiewicz traces, which need to be transitively closed in order to provide full information, e.g., about event precedence, in the form of partial orders. We therefore need a suitable notion of closure for order structures. Again, such a notion for order structures without domain labellings was introduced in [12], as recalled below (note that domain labelling does not play any role in this purely order-theoretic definition).

An *order structure closure* is a mapping given by:

$$\boxed{\mathsf{OS} \xrightarrow{\mathsf{os2ios}} \mathsf{IOS} \qquad \langle \Delta, \rightleftharpoons, \sqsubset, \ell\rangle \mapsto \langle \Delta, \sqsubset^{\circledast} \circ (\rightleftharpoons \cup\ cross^{sym}) \circ \sqsubset^{\circledast}, \sqsubset^{\curlywedge}, \ell\rangle} \quad (4)$$

where $cross = \{\langle x, y\rangle \mid \exists z, z' :\ z \rightleftharpoons z' \ \wedge\ x \sqsubset^{*} z \sqsubset^{*} y \ \wedge\ x \sqsubset^{*} z' \sqsubset^{*} y\}$.

Intuitively, the derived weak causality, $\sqsubset^{\curlywedge}$, captures the fact that weak causality is transitive. The first component of the derived mutex, $\sqsubset^{\circledast} \circ \rightleftharpoons \circ \sqsubset^{\circledast}$, captures the fact that if we have two clusters of simultaneous events, and there is a pair of events in these two clusters which is non-simultaneous, then the same is true of all the pairs of events coming from these clusters (see also Axiom *I5* in Theorem 3). The other component, $\sqsubset^{\circledast} \circ cross^{sym} \circ \sqsubset^{\circledast}$, captures the cross-like propagation of the mutex relationship capture by the diagram below which illustrates the derivation of $\langle x, y\rangle \in cross$ (see also Axiom *I6* in Theorem 3):



We then obtain that order structure closure is the only way in which order structures can be closed to yield invariant order structures.

Proposition 12 os2ios *is the unique structure-closure operator from* OS *to* IOS.

In this way, we have ended our quest for general relational structures corresponding to causal partial orders, and the general notion of invariant order structure and concurrent history as from Theorem 2 and Proposition 12 we obtain:

Theorem 4 *The diagram in Figure 2 commutes.*

5 Step traces and extended causal structures

We now join together the two lines of our discussion, one concerned with generalisation of Mazurkiewicz traces, and the other dealing with extensions of causal partial orders. Throughout this section, $\theta = \langle \Sigma, \mathsf{sim}, \mathsf{seq}\rangle$ is a **fixed** step alphabet.

5.1 Step sequences and order structures

We need to formally establish the correspondence between step sequences from SSEQ_θ and saturated order structures, similarly to the way in which sequences

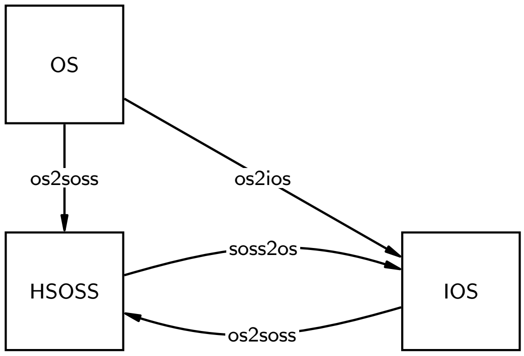


Fig. 2 Correspondence diagram for order structures. where: OS are order structures (Def.3), IOS are invariants (Def.6 & Thm.3), HSOSS are histories (Def.7), os2ios is closure (Eq.(4) & Prop.12), soss2os is intersection (Eq.3 & Thm.2), and os2soss is saturation (Eq.3 & Thm.2).

can be interpreted as total orders. Moreover, we will later be in a position to lift the notion of a trace to the level of SO-structures.

It follows from Proposition 4 that isomorphisms between label-linear relational structures are unique and so we are free to choose the names of the elements that will carry the action names as labels. We therefore focus on order structures whose domains can be seen as a set of events which occurred during an execution of a concurrent system.

Definition 8 (consistent SO-structure) An SO-structure $sos = \langle \Delta, \rightleftharpoons, \sqsubset, \ell \rangle$ is *consistent* with θ if there is a mapping $\Sigma \xrightarrow{\epsilon} \mathbb{N}$ such that $\Delta = \{\langle a,i \rangle \mid a \in \Sigma \wedge 1 \leq i \leq \epsilon(a)\}$ and, for all distinct $\langle a,i\rangle, \langle a,j\rangle, \langle b,k\rangle \in \Delta$, we have $\ell(\langle a,i\rangle) = a$ and:

$$\begin{array}{rcl} \langle a,i\rangle \prec \langle a,j\rangle & \iff & i < j \\ \langle a,i\rangle \sqsubset^{\circledast} \langle b,k\rangle & \implies & \langle a,b\rangle \in \mathsf{sim}\ . \end{array} \quad (5)$$

All SO-structures consistent with θ are denoted by SOS_θ. ⋄

In other words, in a consistent SO-structure, consecutive occurrences of events with the same label are totally ordered, and the labels of events that happen simultaneously are occurrences of actions that can be simultaneous according to θ.

Consistent SO-structures correspond exactly to the step sequences in SSEQ_θ as we now proceed to prove. First we show how such structures can be interpreted as sequences of sets of simultaneous events.

Proposition 13 *Let* $sos = \langle \Delta, \rightleftharpoons, \sqsubset, \ell \rangle \in \mathsf{SOS}_\theta$. *Then there is a unique sequence* $\tau_{sos} = \Delta_1 \ldots \Delta_k$ *such that:*

1. $\Delta_1, \ldots, \Delta_k$ *is a partition of the domain* Δ *satisfying* $\rightleftharpoons\, = \bigcup\{\Delta_i \times \Delta_j \mid i \neq j\}$, $\sqsubset\, = \bigcup\{\Delta_i \times \Delta_j \mid i \leq j\} \setminus id_\Delta$ *and* $\prec\, = \bigcup\{\Delta_i \times \Delta_j \mid i < j\}$.
2. $\Delta_1, \ldots, \Delta_k$ *are the equivalence classes of* $\sqsubset^{\circledast}$.

The unique sequence τ_{sos} in Proposition 13 will be called the *layer decomposition* of *sos*. This decomposition defines through its labeling a step sequence associated with *sos*.

Proposition 14 *Let $\tau_{sos} = \Delta_1 \ldots \Delta_k$ be the layer decomposition of $sos \in \mathsf{SOS}_\theta$. Then, for all $1 \leq i \leq k$, labelling ℓ is injective on Δ_i and $\ell(\Delta_i) \in \mathbb{S}_\theta$.*

Thus a consistent SO-structure corresponds to a sequence of layers, each layer comprising events which can be seen as a valid step according to the alphabet θ. We now take advantage of this observation to establish the full correspondence between SO-structures in SOS_θ and step sequences in SSEQ_θ, using two mappings:

$$\begin{array}{|lr|} \mathsf{SOS}_\theta \xrightarrow{\mathsf{sos2sseq}} \mathsf{SSEQ}_\theta & sos \mapsto \ell(\tau_{sos}) \\ \mathsf{SSEQ}_\theta \xrightarrow{\mathsf{sseq2sos}} \mathsf{SOS}_\theta & u \mapsto \langle \Delta, \rightleftharpoons, \sqsubset, \ell \rangle \end{array} \quad (6)$$

where $\Delta = occ(u)$ and, for all $\alpha, \beta \in occ(u)$ with $pos_u(\alpha) = k$ and $pos_u(\beta) = m$:

$$\begin{array}{lll} \alpha \rightleftharpoons \beta & \text{if} & k \neq m \\ \alpha \sqsubset \beta & \text{if} & k \leq m \ \wedge \ \alpha \neq \beta\,. \end{array} \quad (7)$$

Theorem 5 $\mathsf{SSEQ}_\theta \xrightarrow{\mathsf{sseq2sos}} \mathsf{SOS}_\theta \xrightarrow{\mathsf{sos2sseq}} \mathsf{SSEQ}_\theta$ *are inverse bijections.*

Labelling the layer decomposition of the order structure of a step sequence in SSEQ_θ is the same as listing this step sequence with explicit action occurrences.

Proposition 15 *If $u \in \mathsf{SSEQ}_\theta$ then $\tau_{\mathsf{sseq2sos}(u)} = occseq(u)$.*

Finally, using the bijective correspondence between step sequences and their underlying labelled order structures, we lift the concept of step trace to the level of SO-structures.

Let $sos, sos' \in \mathsf{SOS}_\theta$. Then $sos \mathrel{\dot{\approx}_\theta} sos'$ if $\mathsf{sos2sseq}(sos) \approx_\theta \mathsf{sos2sseq}(sos')$. We denote by $\doteq_\theta$ the reflexive transitive closure of $\dot{\approx}_\theta$.

Definition 9 (order structure trace) An *order structure trace* over θ is an equivalence class of $\doteq_\theta$. All order structure traces will be denoted by OSTR_θ. $\diamond$

We then obtain, by Theorem 5 and the definition of OSTR_θ:

Theorem 6 $\mathsf{STR}_\theta \xrightarrow{\mathsf{sseq2sos}} \mathsf{OSTR}_\theta \xrightarrow{\mathsf{sos2sseq}} \mathsf{STR}_\theta$ *are inverse bijections.*

The above result provides a strong as well as convenient method for moving between the language-theoretic and order-theoretic descriptions of concurrent histories, respectively captured by step traces order structure traces.

5.2 Step sequences and dependence structures

The information assembled in the alphabet θ is sufficient to capture the intrinsic dependencies between events involved in a single step sequence over θ. For this we define the following mapping:

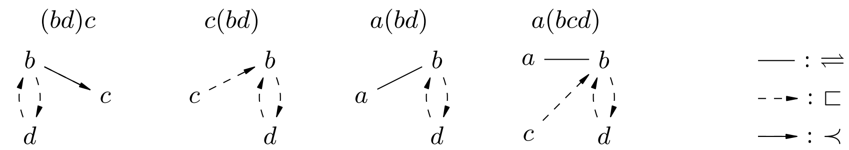
$$\boxed{\mathsf{SSEQ}_\theta \xrightarrow{\mathsf{sseq2os}_\theta} \mathsf{OS} \qquad u \mapsto \langle \Delta, \rightleftharpoons, \sqsubset, \ell \rangle}$$

where $\Delta = occ(u)$, for all $\alpha, \beta \in \Delta$ with $pos_u(\alpha) = k$ and $pos_u(\beta) = m$:

$$\begin{array}{llll} \alpha \rightleftharpoons \beta & \textit{if} \ \langle \ell(\alpha), \ell(\beta)\rangle \notin \mathsf{sim} \cap \mathsf{seq} & (\in \mathsf{ssi} \cup \mathsf{wdp} \cup \mathsf{rig} \cup \mathsf{inl}) & \wedge\ k < m \\ & \textit{or} \ \langle \ell(\alpha), \ell(\beta)\rangle \notin \mathsf{sim} \cap \mathsf{seq}^{-1} & (\in \mathsf{ssi} \cup \mathsf{sse} \cup \mathsf{rig} \cup \mathsf{inl}) & \wedge\ k > m \\ \alpha \sqsubset \beta & \textit{if} \ \langle \ell(\alpha), \ell(\beta)\rangle \notin \mathsf{seq} \cap \mathsf{seq}^{-1} & (\in \mathsf{ssi} \cup \mathsf{sse} \cup \mathsf{wdp} \cup \mathsf{rig}) & \wedge\ k < m \\ & \textit{or} \ \langle \ell(\alpha), \ell(\beta)\rangle \in \mathsf{sim} \setminus \mathsf{seq}^{-1} & (\in \mathsf{ssi} \cup \mathsf{sse}) & \wedge\ k = m \end{array} \tag{8}$$

We refer to $\mathsf{sseq2os}_\theta(u)$ as the *dependence structure* of u. The definition of dependence structure explicitly indicates if two action occurrences are weakly causally related and/or mutual exclusive or neither based on their relative order in the sequence and their mutual relation as given in θ. Consider, e.g., the first line in the definition: two occurrences, that are not in the same step and have labels that cannot be sequentialised when in the same step, are to be connected by the mutex relation. As another example, the last line states that occurrences of two actions are weakly causally related whenever they occur in the same step and a sequentialisation with the second action occurring before the first one is not possible. Note that the definition given above refers also to the semantical relationships between actions as discussed in Section 3.

Example 5 Let θ_0 be as in Example 2. The following are some dependence structures generated from step sequences in SSEQ_{θ_0}:



With the next proposition we establish a number of properties involving dependence structures. In particular, the mapping $\mathsf{sseq2os}_\theta$ is well-defined, and by taking advantage of the additional semantical relationships from Section 3 all possible relationships in a dependence structure can be characterised in a concise way.

Proposition 16 *Let* $u \in \mathsf{SSEQ}_\theta$ *and* $os = \mathsf{sseq2os}_\theta(u) = \langle \Delta, \rightleftharpoons, \sqsubset, \ell \rangle$.

(i) $\rightleftharpoons$ *is symmetric, and both* $\rightleftharpoons$ *and* $\sqsubset$ *are irreflexive.*
(ii) *If* $\alpha, \beta \in \Delta$ *with* $pos_u(\alpha) = k$ *and* $pos_u(\beta) = m$*, then:*

$$\begin{array}{lcll} \alpha \not\sqsubset \beta \wedge \beta \not\sqsubset \alpha \wedge \alpha \not\rightleftharpoons \beta & \Longleftrightarrow & \langle \ell(\alpha), \ell(\beta)\rangle \in \mathsf{con} & \\ \alpha \sqsubset \beta \wedge \beta \sqsubset \alpha \wedge \alpha \not\rightleftharpoons \beta & \Longleftrightarrow & \langle \ell(\alpha), \ell(\beta)\rangle \in \mathsf{ssi} & \wedge\ k = m \\ \alpha \not\sqsubset \beta \wedge \beta \not\sqsubset \alpha \wedge \alpha \rightleftharpoons \beta & \Longleftrightarrow & \langle \ell(\alpha), \ell(\beta)\rangle \in \mathsf{inl} & \wedge\ k \neq m \\ \alpha \sqsubset \beta \wedge \beta \not\sqsubset \alpha \wedge \alpha \not\rightleftharpoons \beta & \Longleftrightarrow & \langle \ell(\alpha), \ell(\beta)\rangle \in \mathsf{sse} & \wedge\ k \leq m \\ \alpha \sqsubset \beta \wedge \beta \not\sqsubset \alpha \wedge \alpha \rightleftharpoons \beta & \Longleftrightarrow & \langle \ell(\alpha), \ell(\beta)\rangle \in \mathsf{ssi} \cup \mathsf{wdp} \cup \mathsf{rig} & \wedge\ k < m \end{array}$$

(iii) *If* $\langle a, i\rangle, \langle a, j\rangle \in \Delta$ *then* $\langle a, i\rangle \prec \langle a, j\rangle \Longleftrightarrow i < j$.
(iv) *If* $\alpha \sqsubset^{\circledast} \beta$ *and* $\alpha \neq \beta$*, then* $pos_u(\alpha) = pos_u(\beta)$ *and* $\langle \ell(\alpha), \ell(\beta)\rangle \in \mathsf{sim}$.
(v) $\mathsf{os2soss}(os) \subseteq \mathsf{SOS}_\theta$.
(vi) $\mathsf{sseq2os}_\theta$ *is a well-defined mapping.*

We can therefore use to kinds of order structures to capture the causal dependencies of action occurrences in the step sequences consistent with θ.

Definition 10 (dependencies between events) The dependence structures and invariant order structures generated by step sequences compatible with the step alphabet θ are as follows:

$$\boxed{\mathsf{OS}_\theta = \mathsf{sseq2os}_\theta(\mathsf{SSEQ}_\theta)} \quad \text{and} \quad \boxed{\mathsf{IOS}_\theta = \mathsf{os2ios}(\mathsf{OS}_\theta)} \qquad \diamond$$

Note 5 Referring to the setup of Mazurkiewicz traces, dependence structures correspond to the dependence graphs of action sequences, and invariant order structures to the underlying causal labelled partial orders.

5.3 Dependence structures and traces

We finally investigate the relationships between dependence structures and step traces. First we show that every step sequence can be generated from its dependence structure, and that equivalent step sequences generate the same dependence structures.

Proposition 17 *Let $u, w \in \mathsf{SSEQ}_\theta$.*

1. $u \in \mathsf{sos2sseq} \circ \mathsf{os2soss} \circ \mathsf{sseq2os}_\theta(u)$.
2. $u \equiv_\theta w$ *implies* $\mathsf{sseq2os}_\theta(u) = \mathsf{sseq2os}_\theta(w)$.

Consequently, we can associate dependence structures with step traces:

$$\boxed{\mathsf{STR}_\theta \xrightarrow{\mathsf{sseq2os}_\theta} \mathsf{OS} \qquad [\![u]\!]_\theta \mapsto \mathsf{sseq2os}_\theta(u)} \tag{9}$$

Then, directly from Proposition 17(1), we obtain:

Proposition 18 $\mathsf{STR}_\theta \xrightarrow{\mathsf{sseq2os}_\theta} \mathsf{OS}$ *is a well-defined mapping.*

Now we turn to the reverse question, namely whether all step sequences defined by a dependence structure are equivalent (and could thus form a step trace). To deal with this we found it convenient to single out steps which cannot be sequentialised.

A *min-step* is $A \in \mathbb{S}_\theta$ such that there are no steps B, C satisfying $A = B \cup C$ and $B \times C \subseteq \mathsf{seq}$. A step sequence $u \in \mathsf{SSEQ}_\theta$ is *thin* if it is composed of min-steps. All thin step sequences are denoted by $\mathsf{SSEQ}_\theta^{thin}$.

Example 6 Let θ_0 be as in Example 2. The step trace $[\![a(bcd)]\!]_{\theta_0}$ contains three thin step sequences: $ac(bd)$, $ca(bd)$ and $c(bd)a$; and three non-thin ones: $a(bcd)$, $(bcd)a$ and $(ac)(bd)$. $\diamond$

Any step sequence can be 'flattened' to yield an equivalent thin step sequence.

Proposition 19 *For every $u \in \mathsf{SSEQ}_\theta$ there is $w \in \mathsf{SSEQ}_\theta^{thin}$ such that $u \equiv_\theta w$.*

We are then ready for the basic result that we need for our proof of the equivalence of all step sequences defined by the dependence structure of a step trace. The proof — presented in the appendix — relies on several auxiliary observations. We start from a thin step sequence and its dependence structure. With the min-steps as 'atomic' building blocks, we first in essence follow the classical approach for Mazurkiewicz traces and their dependence graphs in which the atoms are singleton sets. Recalling that a dependence structure collects all causal (necessary) relations between the min-steps with all other relations being observational and specific to the initial step sequence, we are free to change the order of min-steps as long as we do not violate the invariant causality of the dependence structure. The result is (another) linearisation that is equivalent with the given step sequence. This can be repeated and finally we also combine min-steps into larger steps, still obeying the restrictions of causality imposed by the dependence structure that guarantees equivalence of the thus obtained new step sequence.

Proposition 20 *If* $u \in \mathsf{SSEQ}_\theta$ *and* $w \in \mathsf{sos2sseq} \circ \mathsf{os2soss} \circ \mathsf{sseq2os}_\theta(u)$, *then* $u \equiv_\theta w$.

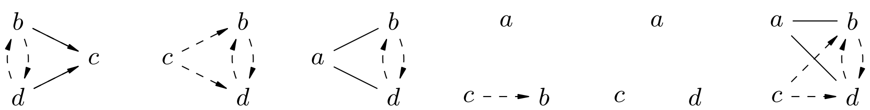
From the results on step traces and order structures, as well as their interrelationships, we can now conclude that we have achieved the main aim of this paper.

Theorem 7 *The diagram in Figure 3 commutes.*

We have therefore obtained a counterpart of the schematic correspondence diagram of Figure 1. In addition, the diagram in Figure 3 provides one more domain, OSTR_θ, which provides a technically convenient bridge between the language-theoretic domain of step traces and the order-theoretic domain of invariant order structures. In Figure 1 — and indeed the standard approach of Mazurkiewicz traces — such a bridge is established 'on-the-fly' by an implicit identification of a sequence of actions with the corresponding labelled total order.

Example 7 Let θ_0 be as in Example 2. The following are some invariant order structures corresponding to step traces in STR_{θ_0}:

$[\![(bd)c]\!]_{\theta_0}$ $\quad [\![c(bd)]\!]_{\theta_0}$ $\quad [\![a(bd)]\!]_{\theta_0}$ $\quad [\![a(bc)]\!]_{\theta_0}$ $\quad [\![a(cd)]\!]_{\theta_0}$ $\quad [\![a(bcd)]\!]_{\theta_0}$



We have demonstrated that step traces can be represented by invariant order structures. A question might therefore arise as to whether such (rather complicated) structures are really necessary, or perhaps a class of simpler order structures would suffice. It turns out that this is not the case.

Proposition 21 *Let os be an order structure with an injective labelling. Then there is a step alphabet* θ *and a step sequence* u *consistent with* θ *such that os is isomorphic to* $\mathsf{sseq2os}_\theta(u)$.

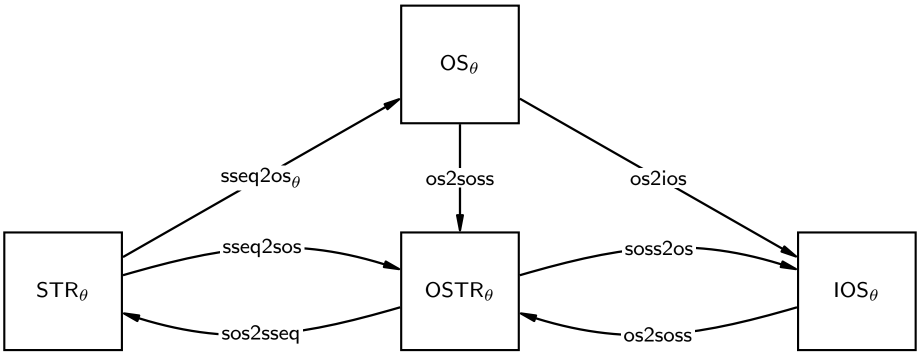
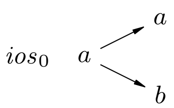


Fig. 3 Correspondence diagram for step traces, where: STR_θ are step traces (Def.2), OSTR_θ are order structure traces (Def.9), OS_θ are dependence structures (Def.10), IOS_θ are invariants (Def.10), os2ios is closure (Eq.(4)), soss2os is intersection (Eq.(3)), os2soss is saturation (Eq.(3)), sseq2os$_\theta$ is derivation of dependence structures (Eq.(9,8)), sos2sseq is transformation of saturated order structures to step sequences (Eq.(6)), and sseq2sos is the reverse transformation (Eq.(6)).

Although in the above we assumed injective labelling, the result we obtained demonstrates that dependence structures of step alphabets can display all the complex patterns involving causal relationships captured by order structures. This is no longer the case if we allow non-injective labellings. Consider, for example, the following IO-structure:



The corresponding history *hsoss* contains three SO-structures sos_i $(i = 1, 2, 3)$ such that $\tau_{sos_1} = aab$, $\tau_{sos_2} = aba$, and $\tau_{sos_3} = a(ab)$. One can see that there is no step alphabet θ such that $\{sos_1, sos_2, sos_3\} \in \mathsf{OSTR}_\theta$ and $ios \in \mathsf{IOS}_\theta$. The intuitive reason is that the first occurrence of a causes b to occur, so a and b are dependent, but the second occurrence of a is concurrent with b, and so a and b are independent. However, in any step alphabet the relationship between a and b is *static* and cannot depend on a specific occurrence of a.

6 Conclusions

In this paper, we have considered an extension of Mazurkiewicz traces taking steps as the smallest units of observation rather than single actions. This extension — being based on a few, light design choices — stays close to the original trace philosophy. We have investigated (labelled) relational structures matching the resulting general step traces and step sequences in the same way as partial orders match Mazurkiewicz traces and total orders match sequences of action occurrences (embodied by the schematic commutative diagram of Figure 1). To represent observational and causal relationships in the behaviours of concurrent systems we used the *order structures* from [12] which are an extension of an idea first proposed in [7,16,26]. Note that a direct predecessor of order structures were the *stratified order structures* (where $\rightleftharpoons$ is included in $\sqsubset$), introduced independently in [7] and [15], and then applied, e.g., in [20,22]. Actually, the approach chosen to lift Mazurkiewicz traces to the setting of step sequences leads to a hierarchy of step alphabets (see also [13], where a slightly different terminology is used) allowing intuitive classifications fitting both established (e.g., comtraces [17] and ST-traces [40,41]), and as yet uninvestigated trace models. In the companion paper [14] these (proper) subclasses are investigated and the order structures representing them identified. The results of our investigations here are captured by the commuting diagram of Figure 3. In essence, with invariant order structures being the most general causal structures representing concurrent histories comprising step sequences (see, eg [12]), it shows that the step traces as proposed in this paper are the most general version of Mazurkiewicz traces in the context of step sequences. We ended our discussions looking at the expressiveness of step traces and concluding that simpler order structures like the ones in [14], would not be sufficient.

When it comes to system models, Mazurkiewicz traces fit elementary net systems [29,36]. To fit the general concurrency paradigm, and hence by the results of this paper, also step traces, the elementary net system model has been extended to include two new kinds of arcs, inhibitor arcs and mutex arcs [24]. As an example, consider Figure 4 showing an elementary net system N extended with an inhibitor and mutex arc. For such Petri nets, the relationships between transitions (actions) can be retrieved directly from the structure of the net, defining a step alphabet θ_N, where the actions are simply the transitions of the net. In this particular example, we have: $\langle a,d\rangle, \langle b,c\rangle \in \mathsf{con}$, $\langle a,b\rangle \in \mathsf{inl}$, $\langle a,c\rangle, \langle b,d\rangle \in \mathsf{rig}$, and $\langle c,d\rangle \in \mathsf{sse}$. Then the set of all step sequences generated by N can be partitioned into step traces conforming to the alphabet θ_N, for example:

$$\begin{aligned} [\![abcd]\!]_{\theta_N} &= \{abcd, ab(cd), bacd, ba(cd), acbd, a(bc)d\} \\ [\![bda]\!]_{\theta_N} &= \{bda, bad, b(ad), abd\}\ . \end{aligned}$$

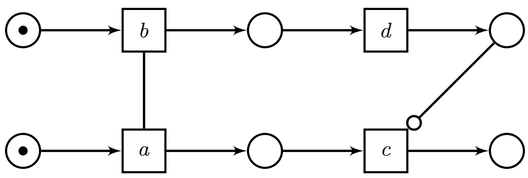


Fig. 4 An elementary net system extended with an inhibitor arc implying that when c is executed the output place of d must be empty, and a mutex arc implying that c and d cannot be executed simultaneously.

It also seems worthwhile to point out differences with some concurrency models from the literature that at first sight might seem related to step traces. First of all, there exist other generalisations of traces. Semi-traces originally introduced as rewriting systems by [4] and later investigated in, e.g., [11,35] are generated by semi-commutations. The rewriting rules that change the order of two adjacent action occurrences can be one-directional, $ab \to ba$, rather than the bi-directional interpretation $ab \leftrightarrow ba$ of Mazurkiewicz independence. This cannot be mimicked with rewriting via steps as done in this paper. Conversely, there *do not exist* partial order models which can deal with 'not later than' situations [16,17]. Other approaches that allow simultaneous executions, i.e., steps, either cannot express any equivalent of 'not later than' [1,37,40], or, as [3,21,41], can equivalently be modelled with the comtraces of [17] (i.e., a special case of the model presented in this paper). In addition, we are not aware of a model that can express a mutex situation represented here by the interleaving equation ($AB = BA$ and $A \cap B = \emptyset$) other than those following [18]. However, the model of [18] does not cover all interesting cases (see [24]), and is a special case of the model considered in this paper. Other extensions of Mazurkiewicz traces consider infinite sequences, leading to complex traces or infinite traces as in, e.g., [5,8]. Finally, it should be noted that the extension of Mazurkiewicz traces discussed in this paper is a *static* one, in contrast to the context or history dependent traces from, e.g., [2,10,25].

Then there are various kinds of events structures based on the seminal work [32]. These structures are single objects describing the full behaviour of a concurrent system, explicitly representing conflict (choice). In contrast, a step trace represents a conflict-free run of a system where mutex is treated as a choice of ordering rather than conflicting behaviour.

We view this paper as a beginning, establishing step traces as the right semantical model fitting the general concurrency paradigm of [16]. Mazurkiewicz approach in which independence, simultaneity and unorderedness are basically the same notion, has been refined and, as a consequence, some of its elegant simple properties, e.g. relating alphabets and equivalence classes, have to be re-investigated. What is also missing is a full investigation of the algebraic, logic and automata-theoretic properties of step traces. For example, it may be the case that the models of [39] or [33] can be suitably extended.

Acknowledgements We are grateful to the reviewers for their useful comments and suggestions. This research was supported by EPSRC (UNCOVER project), the Polish National Science Center (grant No.2013/09/D/ST6/03928), and NSERC of Canada.

References

1. Baldan, P., Busi, N., Corradini, A., Pinna, G.M.: Domain and event structure semantics for Petri nets with read and inhibitor arcs. Information and Computation **323**, 129–189 (2004)
2. Biermann, I., Rozoy, B.: Reliable generalized and context dependent commutation relations. In: TAPSOFT'97, *Lecture Notes in Computer Science*, vol. 1214, pp. 165–176. Springer (1997)
3. Busi, N., Pinna, G.M.: Process semantics for place/transition nets with inhibitor and read arcs. Fundam. Inform. **40**(2-3), 165–197 (1999)
4. Clerbout, M.: Commutations partielles et familles de langages. Ph.D. thesis, University of Lille (1984)
5. Diekert, V., Gastin, P., Petit, A.: Recognizable complex trace languages. In: MFCS 1991, *Lecture Notes in Computer Science*, vol. 520, pp. 131–140. Springer (1991)
6. Diekert, V., Rozenberg, G. (eds.): The Book of Traces. World Scientific (1995)
7. Gaifman, H., Pratt, V.R.: Partial order models of concurrency and the computation of functions. In: LICS, pp. 72–85. IEEE Computer Society (1987)
8. Gastin, P.: Infinite traces. In: Semantics of Systems of Concurrent Processes 1990, *Lecture Notes in Computer Science*, vol. 469, pp. 277–308. Springer (1990)

9. Hoogeboom, H.J., Rozenberg, G.: Dependence graphs. In: V. Diekert, G. Rozenberg (eds.) The Book of Traces, pp. 43–67. World Scientific (1995)
10. Hoogers, P.W., Kleijn, H.C.M., Thiagarajan, P.S.: A trace semantics for Petri nets. Inf. Comput. **117**(1), 98–114 (1995)
11. Hung, D.V., Knuth, E.: Semi-commutations and Petri nets. Theoretical Computer Science **64**, 67–81 (1989)
12. Janicki, R., Kleijn, J., Koutny, M., Mikulski, Ł.: Causal structures for general concurrent behaviours. In: M.S. Szczuka, L. Czaja, M. Kacprzak (eds.) CS&P, *CEUR Workshop Proceedings*, vol. 1032, pp. 193–205. CEUR-WS.org (2013)
13. Janicki, R., Kleijn, J., Koutny, M., Mikulski, Ł.: Generalising traces. TR-CS 1436, Newcastle University (2014)
14. Janicki, R., Kleijn, J., Koutny, M., Mikulski, Ł.: Order structures for subclasses of generalised traces. TR-CS 1437, Newcastle University (2014)
15. Janicki, R., Koutny, M.: Invariants and paradigms of concurrency theory. In: PARLE 1991, *Lecture Notes in Computer Science*, vol. 506, pp. 59–74. Springer (1991)
16. Janicki, R., Koutny, M.: Structure of concurrency. Theor. Comput. Sci. **112**(1), 5–52 (1993)
17. Janicki, R., Koutny, M.: Semantics of inhibitor nets. Inf. Comput. **123**(1), 1–16 (1995)
18. Janicki, R., Le, D.T.M.: Modelling concurrency with comtraces and generalized comtraces. Inf. Comput. **209**(11), 1355–1389 (2011)
19. Janicki, R., Yin, X., Zubkova, N.: Modeling interval order structures with partially commutative monoids. In: CONCUR 2012, *Lecture Notes in Computer Science*, vol. 7454, pp. 425–439. Springer (2012)
20. Juhás, G., Lorenz, R., Mauser, S.: Causal semantics of algebraic Petri nets distinguishing concurrency and synchronicity. Fundam. Inform. **86**(3), 255–298 (2008)
21. Juhás, G., Lorenz, R., Mauser, S.: Complete process semantics of Petri nets. Fundam. Inform. **87**(3-4), 331–365 (2008)
22. Juhás, G., Lorenz, R., Neumair, C.: Synthesis of controlled behavior with modules of signal nets. In: ICATPN 2004, *Lecture Notes in Computer Science*, vol. 3099, pp. 238–257. Springer (2004)
23. Kleijn, J., Koutny, M.: Formal languages and concurrent behaviours. In: G.B. Enguix, M.D. Jiménez-López, C. Martín-Vide (eds.) New Developments in Formal Languages and Applications, *Studies in Computational Intelligence*, vol. 113, pp. 125–182. Springer (2008)
24. Kleijn, J., Koutny, M.: Mutex causality in processes and traces of general elementary nets. Fundam. Inform. **122**(1-2), 119–146 (2013)
25. Kuske, D., Morin, R.: Pomsets for local trace languages - recognizability, logic & Petri nets. In: CONCUR 2000, *Lecture Notes in Computer Science*, vol. 1877, pp. 426–441. Springer (2000)
26. Lamport, L.: The mutual exclusion problem: part I - a theory of interprocess communication. J. ACM **33**(2), 313–326 (1986)
27. Le, D.T.M.: On three alternative characterizations of combined traces. Fundam. Inform. **113**(3-4), 265–293 (2011)
28. Mazurkiewicz, A.: Concurrent program schemes and their interpretations. DAIMI Rep. PB 78, Aarhus University (1977)
29. Mazurkiewicz, A.W.: Trace theory. In: Petri Nets: Central Models and Their Properties, Advances in Petri Nets 1986, Part II, *Lecture Notes in Computer Science*, vol. 255, pp. 279–324. Springer (1987)
30. Mazurkiewicz, A.W.: Basic notions of trace theory. In: J.W. de Bakker, W.P. de Roever, G. Rozenberg (eds.) REX Workshop, *Lecture Notes in Computer Science*, vol. 354, pp. 285–363. Springer (1988)
31. Mikulski, Ł., Koutny, M.: Folded Hasse diagrams of combined traces. Inf. Process. Lett. **114**(4), 208–216 (2014)
32. Nielsen, M., Plotkin, G.D., Winskel, G.: Petri nets, event structures and domains, Part I. TCS **13**, 85–108 (1981)
33. Penczek, W., Kuiper, R.: Traces and logic. In: V. Diekert, G. Rozenberg (eds.) The Book of Traces, pp. 307–390. World Scientific (1995)
34. Pratt, V.: Modeling concurrency with partial orders. International Journal of Parallel Programming **15**(1), 33–71 (1986)
35. Reinhardt, K.: On the synchronization of semi-traces. In: Fundamentals of Computation Theory, *Lecture Notes in Computer Science*, vol. 965, pp. 393–403. Springer (1995)

36. Rozenberg, G., Engelfriet, J.: Elementary net systems. In: Lectures on Petri Nets I: Basic Models, *Lecture Notes in Computer Science*, vol. 1491, pp. 12–121. Springer (1998)
37. Shields, M.: Adequate path expressions. In: G. Kahn (ed.) Semantics of Concurrent Computation, *Lecture Notes in Computer Science*, vol. 70, pp. 249–265. Springer (1979)
38. Szpilrajn, E.: Sur l'extension de l'ordre partiel. Fundam. Math. **16**, 386–389 (1930)
39. Thiagarajan, P., Walukiewicz, I.: An expressively complete linear time temporal logic for mazurkiewicz traces. Information and Computation **179**(2), 230–249 (2002)
40. Vogler, W.: A generalization of trace theory. RAIRO Infornatique théorique et applications **25**(2), 147–156 (1991)
41. Vogler, W.: Partial order semantics and read arcs. Theoretical Computer Science **286**(1), 33–63 (2002)

Proofs of Section 3

Proof of Proposition 1. Follows from the fact that $\mathsf{ser} \subseteq \mathsf{sim}$, and so if we consider an equation $AB =_\psi A \cup B$ with $A, B \in \mathbb{S}_\psi$, we have that $A \cup B \in \mathbb{S}_\psi$.

Proof of Proposition 2. Follows from the fact that for an equation $AB =_\psi A \cup B$ with $A, B \in \mathbb{S}_\psi$, we have that $A \cup B \subseteq \mathsf{sim}$.

Proof of Theorem 1. We first show that the mappings $\mathsf{sa2gca}$ and $\mathsf{gca2sa}$ are well-defined. Let $\psi = \langle \Sigma, \mathsf{sim}, \mathsf{inl}, \mathsf{ser}\rangle \in \Psi$. Then $\mathsf{sa2gca}(\psi) = \langle \Sigma, \mathsf{sim}, \mathsf{inl} \cup \mathsf{ser}\rangle \in \Theta$. Indeed, $\mathsf{inl} \cup \mathsf{ser}$ is clearly irreflexive. Moreover, by $\mathsf{ser} \subseteq \mathsf{sim}$ and $\mathsf{inl} \cap \mathsf{sim} = \emptyset$, $(\mathsf{inl} \cup \mathsf{ser}) \setminus \mathsf{sim} = \mathsf{inl} \setminus \mathsf{sim} = \mathsf{inl}$. Hence $(\mathsf{inl} \cup \mathsf{ser}) \setminus \mathsf{sim}$ is symmetric as inl is.

Now, let $\theta = \langle \Sigma, \mathsf{sim}, \mathsf{seq}\rangle \in \Theta$. Then

$$\mathsf{gca2sa}(\theta) = \langle \Sigma, \mathsf{sim}, (\mathsf{seq} \cap \mathsf{seq}^{-1}) \setminus \mathsf{sim}, \mathsf{seq} \cap \mathsf{sim}\rangle \in \Psi \ .$$

Indeed, $(\mathsf{seq} \cap \mathsf{seq}^{-1}) \setminus \mathsf{sim}$ and $\mathsf{seq} \cap \mathsf{sim}$ are clearly irreflexive, $(\mathsf{seq} \cap \mathsf{seq}^{-1}) \setminus \mathsf{sim}$ is symmetric, and $\mathsf{seq} \cap \mathsf{sim} \subseteq \mathsf{sim}$. Moreover, $((\mathsf{seq} \cap \mathsf{seq}^{-1}) \setminus \mathsf{sim}) \cap \mathsf{sim} = \varnothing$.

To show that the mappings are inverse bijections, we show that

$$\mathsf{gca2sa} \circ \mathsf{sa2gca}(\psi) = \psi \quad \text{and} \quad \mathsf{sa2gca} \circ \mathsf{gca2sa}(\theta) = \theta \ ,$$

for all $\psi = \langle \Sigma, \mathsf{sim}, \mathsf{inl}, \mathsf{ser}\rangle \in \Psi$ and $\theta = \langle \Sigma, \mathsf{sim}, \mathsf{seq}\rangle \in \Theta$. Indeed, we have that

$$\begin{aligned}\mathsf{gca2sa} \circ \mathsf{sa2gca}(\psi) &= \mathsf{gca2sa}(\langle \Sigma, \mathsf{sim}, \mathsf{inl} \cup \mathsf{ser}\rangle)\\ &= \langle \Sigma, \mathsf{sim}, ((\mathsf{inl} \cup \mathsf{ser}) \cap (\mathsf{inl} \cup \mathsf{ser})^{-1}) \setminus \mathsf{sim}, (\mathsf{inl} \cup \mathsf{ser}) \cap \mathsf{sim}\rangle\\ &= \langle \Sigma, \mathsf{sim}, \mathsf{inl}, \mathsf{ser}\rangle \ ,\end{aligned}$$

where the last equality follows from

$$\begin{aligned}&((\mathsf{inl} \cup \mathsf{ser}) \cap (\mathsf{inl} \cup \mathsf{ser})^{-1}) \setminus \mathsf{sim}\\ &\quad = ((\mathsf{inl} \cap \mathsf{inl}^{-1}) \cup (\mathsf{ser} \cap \mathsf{inl}^{-1}) \cup (\mathsf{inl} \cap \mathsf{ser}^{-1}) \cup (\mathsf{ser} \cap \mathsf{ser}^{-1})) \setminus \mathsf{sim}\\ &\quad = (\mathsf{inl} \cap \mathsf{inl}^{-1}) \setminus \mathsf{sim} \cup (\mathsf{ser} \cap \mathsf{ser}^{-1}) \setminus \mathsf{sim} = \mathsf{inl} \cap \mathsf{inl}^{-1} = \mathsf{inl}\end{aligned}$$

and the symmetry of inl, $\mathsf{sim} \cap \mathsf{inl} = \varnothing$, and $\mathsf{ser} \subseteq \mathsf{sim}$, as well as

$$(\mathsf{inl} \cup \mathsf{ser}) \cap \mathsf{sim} = (\mathsf{inl} \cap \mathsf{sim}) \cup (\mathsf{ser} \cap \mathsf{sim}) = \mathsf{ser} \cap \mathsf{sim}$$

and $\mathsf{ser} \subseteq \mathsf{sim}$. We then observe that

$$\begin{aligned}\mathsf{sa2gca} \circ \mathsf{gca2sa}(\theta) &= \mathsf{sa2gca}(\langle \Sigma, \mathsf{sim}, (\mathsf{seq} \cap \mathsf{seq}^{-1}) \setminus \mathsf{sim}, \mathsf{seq} \cap \mathsf{sim}\rangle)\\ &= \langle \Sigma, \mathsf{sim}, ((\mathsf{seq} \cap \mathsf{seq}^{-1}) \setminus \mathsf{sim}) \cup (\mathsf{seq} \cap \mathsf{sim})\rangle = \langle \Sigma, \mathsf{sim}, \mathsf{seq}\rangle \ ,\end{aligned}$$

where the last equality follows from

$$\begin{aligned}((\mathsf{seq} \cap \mathsf{seq}^{-1}) \setminus \mathsf{sim}) \cup (\mathsf{seq} \cap \mathsf{sim}) &= ((\mathsf{seq} \setminus \mathsf{sim}) \cap (\mathsf{seq}^{-1} \setminus \mathsf{sim}))) \cup (\mathsf{seq} \cap \mathsf{sim})\\ &= (\mathsf{seq} \setminus \mathsf{sim}) \cup (\mathsf{seq} \cap \mathsf{sim}) = \mathsf{seq}\end{aligned}$$

and $\mathsf{seq} \setminus \mathsf{sim} = \mathsf{seq} \setminus \mathsf{sim} \cap \mathsf{seq}^{-1} \setminus \mathsf{sim}$ which holds because $\mathsf{seq} \setminus \mathsf{sim}$ is symmetric. To prove that sa2gca and gca2sa are trace-preserving, it suffices to show that $\mathsf{STR}_{\mathsf{sa2gca}(\psi)} = \mathsf{STR}_{\psi}$, for every $\psi = \langle \Sigma, \mathsf{sim}, \mathsf{inl}, \mathsf{ser} \rangle \in \Psi$.

Let $\mathsf{sa2gca}(\psi) = \langle \Sigma, \mathsf{sim}, \mathsf{seq} \rangle$. Then, clearly $\mathbb{S}_{\mathsf{sa2gca}(\psi)} = \mathbb{S}_{\psi}$. Moreover, $\mathsf{seq} \cap \mathsf{sim} = \mathsf{ser} \cap \mathsf{sim} = \mathsf{ser}$ as we have $\mathsf{inl} \cap \mathsf{sim} = \varnothing$ and $\mathsf{ser} \subseteq \mathsf{sim}$, and so the serialisability equations induced by the two alphabets are the same. The interleaving equations are also the same, as we have:

$$\begin{aligned} \mathsf{seq} \cap \mathsf{seq}^{-1} &= (\mathsf{inl} \cup \mathsf{ser}) \cap (\mathsf{inl} \cup \mathsf{ser})^{-1} = (\mathsf{inl} \cup \mathsf{ser}) \cap (\mathsf{inl}^{-1} \cup \mathsf{ser}^{-1}) \\ &= (\mathsf{inl} \cup \mathsf{ser}) \cap (\mathsf{inl} \cup \mathsf{ser}^{-1}) = \mathsf{inl} \cup (\mathsf{ser} \cap \mathsf{ser}^{-1}) \ . \end{aligned}$$

Hence ψ and $\mathsf{sa2gca}(\psi)$ induce the same equations over $\mathbb{S}^*_{\mathsf{sa2gca}(\psi)} = \mathbb{S}^*_{\psi}$. We can therefore conclude that $\mathsf{STR}_{\mathsf{sa2gca}(\psi)} = \mathsf{STR}_{\psi}$.

Proofs of Section 4

Proof of Proposition 3. Let $rs = \langle \Delta, \rightleftharpoons, \sqsubset, \ell \rangle$ be a separable label-ordered relational structure. Suppose that $a \in \Sigma$ and $x, y, z \in \ell^{-1}(a)$ and $x \prec z \prec y$. First, we observe that $x \neq y$ since otherwise we would obtain a contradiction with the separability of rs. Hence by rs being label-ordered, we have $x \prec^{sym} y$. If $y \prec x$, we again obtain a contradiction with the separability of rs. Hence $x \prec y$.

Proof of Proposition 4. Let $rs \sim_\kappa rs'$ be isomorphic label-linear relational structures, and let $a \in \Sigma$. By the label-preservation of κ, κ is a bijection between $\ell_{rs}^{-1}(a)$ and $\ell_{rs'}^{-1}(a)$. Hence, by the label-linearity of rs, κ restricted to $\ell_{rs}^{-1}(a)$ is unique.

Proof of Proposition 5. ***(1)*** By Eq.(1), $rs' \lhd \mathsf{cls}(rs')$. Moreover, $rs' \lhd rs'$ and so, by Eq.(2), $\mathsf{cls}(rs') \lhd rs'$. Hence $rs' \lhd \mathsf{cls}(rs') \lhd rs'$, and so $\mathsf{cls}(rs') = rs'$.

(2) Let $rs'' \in \mathsf{F}'$. We need to show that $rs \lhd rs''$ iff $\mathsf{cls}(rs) \lhd rs''$. The left-to-right implication follows from Eq.(2). Moreover, the right-to-left implication follows from Eq.(1).

Proof of Proposition 7. Clearly, $\prec^a_{\bigcap rss} \subseteq \prec^a_{rs}$. Moreover, $\prec^a_{rs} \subseteq \prec^a_{\bigcap rss}$ as otherwise $\prec^a_{\bigcap rss}$ would not be a total order relation (note that $\prec^a_{rs}$ is a total order relation).

Proof of Proposition 8. ($\Longrightarrow$) Follows from the definition of isomorphism between rss and rss'.

($\Longleftarrow$) First, we observe that all relational structures within rss (and also within rss') are non-isomorphic. Indeed, suppose that $rs \sim_\kappa rs'$, for some $rs, rs' \in rss$. Then, by Proposition 7, we have that κ is the identity on Δ_{rss}. Hence $rs = rs'$. It therefore follows that there is a unique bijection $rss \xrightarrow{\phi} rss'$ relating isomorphic relational structures.

Suppose now that $rs \sim_\kappa \phi(rs)$ and $rs' \sim_{\kappa'} \phi(rs')$. By Proposition 4, both κ and κ' are unique isomorphisms. It then follows from Proposition 7 that $\kappa|_{\ell^{-1}(a)} = \kappa'|_{\ell^{-1}(a)}$, for every $a \in \Sigma$. Hence $\kappa = \kappa'$.

Proof of Proposition 10. Let $sos = \langle \Delta, \rightleftharpoons, \sqsubset, \ell \rangle \in \mathsf{SOS}$. First we show that sos is separable:

- Suppose that $x \sqsubset x$. Then $x \sqsubset x \sqsubset x$ and so, by Axiom *L3*, $x \rightleftharpoons x$ which produces a contradiction. Hence $\sqsubset$ is irreflexive. Therefore, by Axiom *L2*, $\rightleftharpoons$ is also irreflexive.
- Suppose that $x \rightleftharpoons y$ and $x \neq y$. Then, by Axiom *L3*, we have $x \sqsubset y \sqsubset x$ and thus also $y \sqsubset x \sqsubset y$ which in turn implies $y \rightleftharpoons x$. Hence $\rightleftharpoons$ is symmetric.
- Suppose that $x \sqsubset^{\circledast} y$. If $x = y$ then, by the irreflexivity of $\rightleftharpoons$, we have $x \not\rightleftharpoons y$. If $x \neq y$ then, by repeated application of Axiom *L1*, $x \sqsubset y \sqsubset x$. Hence, by Axiom *L3*, $x \rightleftharpoons y$ and so we can conclude that $\rightleftharpoons \cap \sqsubset^{\circledast} = \varnothing$.

As a result, sos is separable. Moreover, sos is label-ordered. Indeed, suppose that $x \neq y$ and $\ell(x) = \ell(y)$. Then, by Axiom *L4*, $x \not\rightleftharpoons y$ and so, by Axiom *L2*, we have $x \sqsubset^{sym} y$. Thus $x \prec^{sym} y$.

We can therefore conclude that $sos \in \mathsf{OS}$. To show that $sos \in \mathsf{SOS}$, suppose that $os \neq sos$ is an order structure such that $sos \lhd os$. Then there must exist $x, y \in \Delta$ such that one of the following holds:

- $x \rightleftharpoons_{os} y$ and $x \not\rightleftharpoons y$. Since $\rightleftharpoons_{os}$ is irreflexive, $x \neq y$. Hence, by Axiom *L3*, $x \sqsubset y \sqsubset x$. Therefore, by $sos \lhd os$, $x \sqsubset^{\circledast}_{os} y$ which, together with $x \rightleftharpoons_{os} y$, contradicts the separability of os.
- $x \sqsubset_{os} y$ and $x \not\sqsubset y$. Since $\sqsubset_{os}$ is irreflexive, $x \neq y$. Hence, by Axiom *L3*, we have $x \rightleftharpoons y$. Thus, by Axiom *L2* and $x \not\sqsubset y$, we obtain $y \sqsubset x$. Therefore, by $sos \lhd os$, $x \sqsubset^{\circledast}_{os} y$ and $x \rightleftharpoons_{os} y$, contradicting the separability of os.

Since in both cases we obtained a contradiction, sos is a saturated order structure.

Conversely, let $os = \langle \Delta, \rightleftharpoons, \sqsubset, \ell \rangle \in \mathsf{SOS}$. We first show that if $x \neq y$ then:

(a) $x \not\rightleftharpoons y$ implies $x \sqsubset^{+} y \sqsubset^{+} x$.
(b) $x \not\sqsubset y$ implies $y \sqsubset^{+} x$ and $x \rightleftharpoons y$.

(a) We first observe that $y \not\rightleftharpoons x$, as $\rightleftharpoons$ is symmetric. We then consider a relational structure os' obtained from os by adding the pair $\langle x,y \rangle$ to $\rightleftharpoons$. Since $os' \neq os$ and $os \lhd os'$, it follows from $os \in \mathsf{SOS}$ that $os' \notin \mathsf{OS}$. We then observe that in such a case $\langle x,y \rangle$ must belong to $\rightleftharpoons_{os'} \cap \sqsubset^{\circledast}_{os'}$. Hence, by $\sqsubset^{\circledast}_{os'} = \sqsubset^{\circledast}$, we obtain that $x \sqsubset^{+} y \sqsubset^{+} x$.

(b) We consider a relational structure os' obtained from os by adding the pair $\langle x,y \rangle$ to $\sqsubset$. As in the case of *(a)*, $os' \notin \mathsf{OS}$. We then observe that in such a case there is a pair $\langle w,u \rangle$ belonging to $\rightleftharpoons_{os'} \cap \sqsubset^{\circledast}_{os'}$. Clearly, $w \rightleftharpoons u$ and the only way that $w \sqsubset^{\circledast}_{os'} u$ holds is that we created a cycle through adding $\langle x,y \rangle$ to $\sqsubset$. Hence we must have had $y \sqsubset^{+} x$. Suppose that $x \not\rightleftharpoons y$. Then, by *(a)*, $x \sqsubset^{\circledast} y$ and so $w \sqsubset^{\circledast} u$ which produces a contradiction with the separability of os. Hence $x \rightleftharpoons y$, and so (b) holds.

We will now show that os is an so-structure, by checking the satisfaction of the defining conditions Axioms *L1*–*L4*:

- Suppose that $x \neq y$ and $x \sqsubset z \sqsubset y$ and $x \not\sqsubset y$. Then, by (b), $y \sqsubset^{+} x$ and $x \rightleftharpoons y$. Thus $y \sqsubset^{+} x \sqsubset z \sqsubset y$, and so $\langle x,y \rangle$ belongs to $\rightleftharpoons \cap \sqsubset^{\circledast}$, contradicting the separability of os. As a result, os satisfies Axiom *L1*.
- Suppose that $x \rightleftharpoons y$ and $x \not\sqsubset^{sym} y$ (i.e., $x \not\sqsubset y$ and $y \not\sqsubset x$). Since $\rightleftharpoons$ is irreflexive, $x \neq y$. Then, by (a), $y \sqsubset^{+} x$ and $x \sqsubset^{+} y$. Thus $y \sqsubset^{+} x \sqsubset^{+} y$, and so $\langle x,y \rangle$ belongs to $\rightleftharpoons \cap \sqsubset^{\circledast}$, contradicting the separability of os. As a result, os satisfies Axiom *L2*.
- Suppose first that $x \neq y$ and $x \not\rightleftharpoons y$. Then, by (a), $x \sqsubset^{+} y \sqsubset^{+} x$, and so, by an already demonstrated Axiom *L1*, $x \sqsubset y \sqsubset x$. Conversely, suppose that $x \sqsubset y \sqsubset x$. Then, by the irreflexivity of $\sqsubset$, we have $x \neq y$, and, by $\rightleftharpoons \cap \sqsubset^{\circledast} = \varnothing$, we have $x \not\rightleftharpoons y$. As a result, os satisfies Axiom *L3*.
- Suppose that $x \neq y$, $\ell(x) = \ell(y)$, and $x \not\rightleftharpoons y$. Then $x \not\prec^{sym} y$, contradicting the label-orderedness of os. As a result, os satisfies Axiom *L4*.

Hence we can conclude that $os \in \mathsf{SOS}$.

Proof of Proposition 11. Suppose that $soss \in \mathsf{SOSS}$ and $os = \mathsf{soss2os}(soss) = \bigcap soss$. Then os is separable by Proposition 6(2), and its label-orderedness follows from the definitions.

Suppose now that $os \in \mathsf{OS}$ and $soss = \mathsf{os2soss}(os) = \mathsf{ext}(os) \cap \mathsf{SOS}$. From Prop.7 and Th.3 in [12], it follows that $soss \neq \varnothing$. Clearly, $soss$ is label-ordered and consistent due to the definition of $\mathsf{ext}(os)$.

Proof of Theorem 2. Suppose that $ios \in \mathsf{IOS}$ and $soss = \mathsf{os2soss}(ios)$. Then, by Definition 6, $ios = \mathsf{soss2os} \circ \mathsf{os2soss}(ios)$. Hence $soss = \mathsf{os2soss}(ios) = \mathsf{os2soss} \circ \mathsf{soss2os} \circ \mathsf{os2soss}(ios) = \mathsf{os2soss} \circ \mathsf{soss2os}(soss)$ and so, by Definition 7, $soss \in \mathsf{HSOSS}$.

Suppose now that $hsoss \in \mathsf{HSOSS}$ and $os = \mathsf{soss2os}(hsoss)$. Then, by Definition 7, $\mathsf{os2soss}(os) = hsoss$. Hence $os = \mathsf{soss2os} \circ \mathsf{os2soss}(os)$ and $os \in \mathsf{IOS}$.

Proof of Theorem 3. Let LGMOS be the set of all *labelled generalised mutex order structures* defined as the set of all relational structures $lgmos$ satisfying the axioms in Theorem 3. Then, from Props.7 & 8 and Thm.3 in [12], we obtain that:

Lemma 1

1. *If* $lgmos \in \mathsf{LGMOS}$, *then* $\mathsf{os2soss}(lgmos) \neq \varnothing$ *and* $lgmos = \mathsf{soss2os} \circ \mathsf{os2soss}(lgmos)$.
2. $\mathsf{os2ios}$ *is a structure-closure operator from* OS *to* LGMOS. ◇

$\mathsf{LGMOS} \subseteq \mathsf{IOS}$ holds by Definition 6 and Proposition 1(1).

Suppose that $ios \in \mathsf{IOS}$. Let $lgmos = \mathsf{os2ios}(ios) \in \mathsf{IOS}$. By Propositions 5 and 1(2), $\mathsf{os2soss}(ios) = \mathsf{os2soss}(lgmos)$. Hence $\mathsf{soss2os} \circ \mathsf{os2soss}(ios) = \mathsf{soss2os} \circ \mathsf{os2soss}(lgmos)$ and so, by Definition 6 and Proposition 1(1), $ios = lgmos \in \mathsf{LGMOS}$. Thus $\mathsf{IOS} \subseteq \mathsf{LGMOS}$.

Proof of Proposition 12. By Proposition 1(2) and Theorem 3, $\mathsf{os2ios}$ is a structure-closure operator from OS to IOS.

Suppose $\mathsf{OS} \xrightarrow{\mathsf{cls}} \mathsf{IOS}$ is a structure-closure operator. Let $os \in \mathsf{OS}$. Then $\mathsf{os2soss}(\mathsf{cls}(os)) = \mathsf{os2soss}(os) = \mathsf{os2soss}(\mathsf{os2ios}(os))$, by Proposition 5($ii$) and Proposition 1(2). Hence, by $\mathsf{cls}(os) \in \mathsf{IOS}$ and $\mathsf{os2ios}(os) \in \mathsf{IOS}$, we obtain $\mathsf{cls}(os) = \mathsf{os2ios}(os)$.

Proofs of Section 5

Proof of Proposition 13. Let $\mathcal{X}$ be the set of equivalence classes of $\sqsubset^{\circledast}$. For distinct $X, Y \in \mathcal{X}$, we define $X \dot{\rightleftharpoons} Y$ and $X \dot{\sqsubset} Y$ if, respectively, $(X \times Y) \cap \rightleftharpoons \neq \varnothing$ and $(X \times Y) \cap \sqsubset \neq \varnothing$. We then show that, for distinct $X, Y \in \mathcal{X}$, we have the following:

$$
\begin{array}{llll}
(i) & X \dot{\sqsubset} Y \implies X \times Y \subseteq \sqsubset & (ii) & X \neq Y \implies X \times Y \subseteq \rightleftharpoons \\
(iii) & X \dot{\sqsubset} Y \implies \neg Y \dot{\sqsubset} X & (iv) & X \neq Y \implies X \dot{\sqsubset}^{sym} Y .
\end{array}
$$

Let $\alpha \in X$ and $\beta \in Y$. Since $X \neq Y$, also $\alpha \neq \beta$.

(i) If $X \dot{\sqsubset} Y$, then there exist $\gamma \in X$ and $\delta \in Y$ such that $\gamma \sqsubset \delta$ which together with $\alpha \neq \beta$ implies by Axiom *L1* that $\alpha \sqsubset \beta$.

(ii) If $\alpha \not\rightleftharpoons \beta$ then $\alpha \neq \beta$ implies, by Axiom *L3*, that $\alpha \sqsubset \beta \sqsubset \alpha$, a contradiction.

(iii) Follows from the maximality of $\sqsubset^{\circledast}$.

(iv) We have $\alpha \neq \beta$. If $\alpha \not\rightleftharpoons \beta$ then, by Axiom *L3*, $\alpha \sqsubset \beta \sqsubset \alpha$. Hence $X \dot{\sqsubset} Y \dot{\sqsubset} X$ which contradicts (iii). Thus we have $\alpha \rightleftharpoons \beta$ and so $\alpha \sqsubset^{sym} \beta$, by Axiom *L2*.

Now define $\dot{\prec} = \dot{\sqsubset} \cap \dot{\rightleftharpoons}$. From what we have just shown it follows that $\dot{\prec}$ is a total order relation over $\mathcal{X}$. Moreover, the order in which the equivalence classes of $\sqsubset^{\circledast}$ are ordered by $\dot{\prec}$ gives the desired sequence and verifies its uniqueness.

Proof of Proposition 14. Let $i \leq k$ and suppose that $\alpha, \beta \in \Delta_i$, $\alpha \neq \beta$ and $\ell(\alpha) = \ell(\beta)$. Then, by Axiom *L4*, $\alpha \rightleftharpoons \beta$. Hence $\alpha \not\sqsubset^{\circledast} \beta$ as, by Proposition 10, sos is an order structure, and so it is separable. We therefore obtained a contradiction with Proposition 13(2).

The second part follows from Proposition 13(2) and Eq.(5).

Proof of Theorem 5. We first show that the mappings $\mathsf{sos2sseq}$ and $\mathsf{sseq2sos}$ are well-defined.

The first part follows from Proposition 14. To show the second part, we proceed as follows.

Suppose that $u \in \mathsf{SSEQ}_\theta$ and $sos = \mathsf{sseq2sos}(u) = \langle \Delta, \rightleftharpoons, \sqsubset, \ell \rangle$. First we demonstrate that $sos \in \mathsf{SOS}$ by showing that the Axioms *L1*–*L4* hold.

Axiom *L1* : Suppose that $\alpha \neq \beta$ and $\alpha \sqsubset \gamma \sqsubset \beta$. By Eq.(7), we have $pos_u(\alpha) \leq pos_u(\gamma) \leq pos_u(\beta)$. Hence $pos_u(\alpha) \leq pos_u(\beta)$ and so, by Eq.(7), $\alpha \sqsubset \beta$.

Axiom *L2* : Suppose that $\alpha \rightleftharpoons \beta$. By Eq.(7), we have $pos_u(\alpha) \neq pos_u(\beta)$ and so also $\alpha \neq \beta$. Hence, by Eq.(7), $\alpha \sqsubset^{sym} \beta$.

Axiom *L3* : Suppose that $\alpha \neq \beta$ and $\alpha \not\rightleftharpoons \beta$. Then, by Eq.(7), $pos_u(\alpha) = pos_u(\beta)$. Hence, by Eq.(7), $\alpha \sqsubset \beta \sqsubset \alpha$.
Conversely, suppose that $\alpha \sqsubset \beta \sqsubset \alpha$. Then, by Eq.(7), $pos_u(\alpha) = pos_u(\beta)$ and $\alpha \neq \beta$. Moreover, by Eq.(7), $\alpha \not\rightleftharpoons \beta$.

Axiom *L4* : Suppose that $\alpha \neq \beta$ and $\ell(\alpha) = \ell(\beta)$. Then $pos_u(\alpha) \neq pos_u(\beta)$ and so, by Eq.(7), $\alpha \rightleftharpoons \beta$.

As a result, $sos \in \mathsf{SOS}$.

Suppose now that $\alpha = \langle a, i \rangle \in \Delta$ and $\beta = \langle a, j \rangle \in \Delta$, where $i \neq j$. Then $i < j \iff pos_u(\alpha) = pos_u(\beta)$. Hence, by Eq.(7), the first part of Eq.(5) holds.

Finally, suppose that $\alpha = \langle a, i \rangle \in \Delta$ and $\beta = \langle b, k \rangle \in \Delta$ are such that $\alpha \sqsubset^{\circledast} \beta$ and $\alpha \neq \beta$. Then, by Eq.(7), $pos_u(\alpha) = pos_u(\beta)$. Hence, by $u \in \mathsf{SSEQ}_\theta$, we have $\langle a, b \rangle \in \mathsf{sim}$, and so the second part of Eq.(5) holds.
As a result, $sos \in \mathsf{SOS}_\theta$. Hence both mappings are well-defined.

Suppose now that $u \in \mathsf{SSEQ}_\theta$. By Proposition 15, $\tau_{\mathsf{sseq2sos}(u)} = occseq(u)$. Hence we obtain $\mathsf{sos2sseq}(\mathsf{sseq2sos}(u)) = \ell(\tau_{\mathsf{sseq2sos}(u)}) = \ell(occseq(u)) = u$.

Proof of Proposition 15. Suppose that $occseq(u) = \Delta_1 \dots \Delta_k$ and $sos = \mathsf{sseq2sos}(u) = \langle \Delta, \rightleftharpoons, \sqsubset, \ell \rangle$. Clearly, $\Delta_1, \dots, \Delta_k$ is a partition of Δ. Moreover, from Eq.(7) it follows that $\rightleftharpoons = \bigcup\{\Delta_i \times \Delta_j \mid i \neq j\}$, $\sqsubset = \bigcup\{\Delta_i \times \Delta_j \mid i \leq j\} \setminus id_\Delta$ and $\prec = \bigcup\{\Delta_i \times \Delta_j \mid i < j\}$. Hence, by Proposition 13(1), $\tau_{\mathsf{sseq2sos}(u)} = occseq(u)$.

Proof of Proposition 16. ***(i):*** $\rightleftharpoons$ is symmetric by $(\mathsf{sim} \setminus \mathsf{seq}^{-1})^{-1} = \mathsf{sim}^{-1} \setminus \mathsf{seq} = \mathsf{sim} \setminus \mathsf{seq}$ and Eq.(8). Clearly, it is also irreflexive by Eq.(8). Also, by Eq.(8) and the fact that $\mathsf{sim} \setminus \mathsf{seq}^{-1}$ is irreflexive, $\sqsubset$ is irreflexive.

(ii): Follows directly from Eq.(8).

(iii): Clearly, $\langle a, i \rangle \not\rightleftharpoons \langle a, j \rangle$ for $i = j$, and so without loss of generality we can assume $i < j$ and $pos_u(\langle a, i \rangle) < pos_u(\langle a, j \rangle)$.

Then, by Eq.(8) and $\langle a, a \rangle \notin (\mathsf{sim} \cap \mathsf{seq}) \cup (\mathsf{sim} \cap \mathsf{seq}^{-1})$, $\langle a, i \rangle \rightleftharpoons \langle a, j \rangle \rightleftharpoons \langle a, i \rangle$. Moreover, $\langle a, a \rangle \notin \mathsf{seq} \cap \mathsf{seq}^{-1}$ and so, by Eq.(8), $\langle a, i \rangle \sqsubset \langle a, j \rangle$. We then observe that $\langle a, j \rangle \sqsubset \langle a, i \rangle$ is impossible by Eq.(8) and $pos_u(\langle a, i \rangle) < pos_u(\langle a, j \rangle)$.

(iv): By Eq.(8), $\alpha \sqsubset^{\circledast} \beta$ implies $pos_u(\alpha) = pos_u(\beta)$. This and $\alpha \neq \beta$ means that $\ell(\alpha) \neq \ell(\beta)$. Hence $\langle \ell(\alpha), \ell(\beta) \rangle \in \mathsf{sim}$ since $u \in \mathsf{SSEQ}_\theta$.

(v): The first part of Eq.(5) follows from *(iii)*, and the second from *(iv)*.

(vi): We need to show that *os* is label-linear and separable. The former follows from *(iii)*. Moreover, if $\alpha \sqsubset^{\circledast} \beta$ and $\alpha \neq \beta$ then, by *(iv)*, $pos_u(\alpha) = pos_u(\beta)$. Hence, by Eq.(8), $\alpha \not\rightleftharpoons \beta$ and so *os* is separable.

Proof of Proposition 17. (1) Let $os = \mathsf{sseq2os}_\theta(u) = \langle \Delta, \rightleftharpoons, \sqsubset, \ell \rangle$. By Theorem 5, it suffices to show that $sos = \mathsf{sseq2sos}(u)$ belongs to $\mathsf{os2soss}(os)$.

Thus we prove that *sos* is a saturated version of *os*. Suppose that $\alpha \rightleftharpoons \beta$. Then, by Eq.(8), $pos_u(\alpha) \neq pos_u(\beta)$. Hence $\alpha \rightleftharpoons_{sos} \beta$. Next suppose that $\alpha \sqsubset \beta$. Then, by Eq.(8), $pos_u(\alpha) \leq pos_u(\beta)$. Hence $\alpha \sqsubset_{sos} \beta$. As a result, $sos \in \mathsf{os2soss}(os)$.

(2) Let $\mathsf{sseq2os}_\theta(u) = \langle \Delta, \rightleftharpoons, \sqsubset, \ell \rangle$ and $\mathsf{sseq2os}_\theta(w) = \langle \Delta, \rightleftharpoons', \sqsubset', \ell \rangle$. It suffices to show the result in the following two cases.

Case 1: $u = AB$, $w = BA$ and $A \times B \subseteq \mathsf{seq} \cap \mathsf{seq}^{-1}$. Then, by seq being irreflexive, we have that $occseq(u) = \Delta_1 \Delta_2$ and $occseq(w) = \Delta_2 \Delta_1$, for some Δ_1 and Δ_2. Clearly, $\rightleftharpoons = \rightleftharpoons'$ as $(\mathsf{sim} \cap \mathsf{seq})^{-1} = \mathsf{sim} \cap \mathsf{seq}^{-1}$. Moreover, $\sqsubset = \sqsubset'$ as the following holds, by Eq.(8) and $A \times B \subseteq \mathsf{seq} \cap \mathsf{seq}^{-1}$:

$$((\Delta_1 \times \Delta_2) \cup (\Delta_2 \times \Delta_1)) \cap \sqsubset = ((\Delta_1 \times \Delta_2) \cup (\Delta_2 \times \Delta_1)) \cap \sqsubset' = \varnothing .$$

Case 2: $u = AB$, $w = A \cup B$ and $A \times B \subseteq \mathsf{seq}$. Then, by seq being irreflexive, we have that $occseq(u) = \Delta_1 \Delta_2$ and $occseq(w) = \Delta_1 \uplus \Delta_2$, for some Δ_1 and Δ_2. We then have $\rightleftharpoons = \rightleftharpoons' = \varnothing$ as $A \times B \subseteq \mathsf{sim} \cap \mathsf{seq}$.

Suppose that $\alpha \in \Delta_1$ and $\beta \in \Delta_2$. Then $\alpha \sqsubset \beta$ iff $\langle \ell(\alpha), \ell(\beta) \rangle \in \mathsf{sim} \setminus \mathsf{seq}^{-1}$. Moreover, $\alpha \sqsubset' \beta$ iff $\langle \ell(\alpha), \ell(\beta) \rangle \notin \mathsf{seq} \cap \mathsf{seq}^{-1}$ iff $\langle \ell(\alpha), \ell(\beta) \rangle \in \mathsf{sim} \setminus \mathsf{seq}^{-1}$ (since $\langle \ell(\alpha), \ell(\beta) \rangle \in \mathsf{sim} \cap \mathsf{seq}$). Suppose now that $\alpha \in \Delta_2$ and $\beta \in \Delta_1$, and so $\langle \ell(\beta), \ell(\alpha) \rangle \in \mathsf{seq}$. Then $\alpha \not\sqsubset' \beta$, by Eq.(8). If $\alpha \sqsubset \beta$ then $\langle \ell(\alpha), \ell(\beta) \rangle \in \mathsf{sim} \setminus \mathsf{seq}^{-1}$, contradicting $\langle \ell(\beta), \ell(\alpha) \rangle \in \mathsf{seq}$.
As a result, $\sqsubset = \sqsubset'$.

Proof of Proposition 19. Let $w = A_1 \dots A_k$ be a longest step sequence such that $u \equiv_\theta w$. Suppose that w is not thin, and A_i is not a min-step, for some $i \leq k$. This means that there are steps B, C such that $A_i = B \uplus C$ and $B \times C \subseteq \mathsf{seq} \cap \mathsf{sim}$. Hence $w \approx_\theta A_1 \dots A_{i-1} B C A_{i+1} \dots A_k$, contradicting the choice of w.

Proof of Proposition 20. We start by defining an auxiliary notion and a result.

A *linearisation* of an acyclic binary relation $\ll$ over a finite set X is any enumeration $u = x_1 \dots x_k$ of the elements of X such that $x_i \ll x_j$ implies $i < j$, for all $i, j \leq k$. Furthermore, we write $u \bowtie w$ if $w = x_1 \dots x_{i-1} x_{i+1} x_i x_{i+2} \dots x_k$, where x_i and x_{i+1} are such that $x_i \not\ll x_{i+1} \not\ll x_i$.

Lemma 2 *If u and w are linearisations of $\ll$ then $u \bowtie^* w$.*

Proof We proceed by induction on $|X|$. In the base case, $|X| = 0$, we have that both u and v are the empty enumeration. In the inductive case, $|X| > 0$, we proceed as follows.

Since X is nonempty and finite, and $\ll$ acyclic, there is an $x \in X$ such that there is no $y \in X' = X \setminus \{x\}$ such that $y \ll x$. We now observe that there is a u' such that $u \bowtie^* xu'$. Indeed, suppose that $u = y_1 \dots y_m x u''$. Then, for all $i \leq m$, we have $y_i \not\ll x$ (by the choice of x), and $x \not\ll y_i$ (by u being a linearisation of $\ll$). Hence $u \bowtie^* xy_1 \dots y_m u''$. Similarly, there is w' such that $w \bowtie^* xw'$. We now observe that u' and w' are linearisations of $\ll' = \ll \cap (X' \times X')$. Hence, by the induction hypothesis, $u' \bowtie^* w'$. As a consequence, $xu' \bowtie^* xw'$ and so $u \bowtie^* xu' \bowtie^* xw' \bowtie^* w$. □

Let $u = A_1 \dots A_k$ and $occseq(u) = \Delta_1 \dots \Delta_k$. Moreover, let $\mathcal{X}$ be the set of all equivalence classes of $\sqsubset^{\circledast}$, and $\ll$ be a binary relation over $\mathcal{X}$ such that $X \ll Y$ if $(X \times Y) \cap \rightleftharpoons \neq \varnothing$ and $(X \times Y) \cap \sqsubset \neq \varnothing$.

Lemma 3 *$\ll$ is an acyclic relation, and $occseq(u)$ is a linearisation of $\ll$.*

Proof The first part is obvious, and the second follows from Eq.(7). □

Lemma 4 *$(X \times X) \cap \rightleftharpoons = \varnothing$, for every $X \in \mathcal{X}$.*

Proof Follows from os being an order structure (and its separability). □

Lemma 5 *If ξ is a linearisation of $\ll$ then $sos_\xi = \mathsf{sseq2sos}(\ell(\xi)) \in \mathsf{os2soss}(os)$ and $\tau_{sos_\xi} = \xi$.*

Proof Follows from Lemmata 3 and 4. □

Lemma 6 *$\mathcal{X} = \{\Delta_1, \dots, \Delta_k\}$.*

Proof Consider A_i and Δ_i. Since A_i is a min-step, the graph of the relation $(A_i \times A_i) \setminus \mathsf{seq}$ over A_i is strongly connected. Hence, by Eq.(8), the graph of $\sqsubset$ restricted to the nodes of Δ_i is also strongly connected. Suppose that $\alpha \in \Delta \setminus \Delta_i$ and $\beta \in \Delta_i$ are such that $\alpha \sqsubset^{\circledast} \beta$. Then, by Eq.(8), $pos_u(\alpha) = pos_u(\beta)$ and so $\alpha \in \Delta_i$, a contradiction. It therefore follows that $\Delta_i \in \mathcal{X}$.

The result follows as both $\Delta_1, \dots, \Delta_k$ and $\mathcal{X}$ are partitions of Δ. □

Lemma 7 *Let $sos' \in \mathsf{os2soss}(os)$, $\tau_{sos'} = \Phi_1 \dots \Phi_m$ and $j \leq m$.*

(i) *Φ_j is the union of some Δ_i's.*
(ii) *$(\Phi_j \times \Phi_j) \cap \rightleftharpoons = \varnothing$.*
(iii) *If there is no $sos'' \in \mathsf{os2soss}(os)$ such that $sos' \dot{\approx}_\theta sos''$ and the length of $\tau_{sos''}$ is greater than m, then $\tau_{sos'}$ is a linearisation of $\ll$.*
(iv) *If X and Y are distinct elements of $\mathcal{X}$ satisfying $X \not\ll Y$ and $Y \not\ll X$, then $\ell(X) \times \ell(Y) \subseteq \mathsf{seq} \cap \mathsf{seq}^{-1}$.*
(v) *If ξ is a linearisation of $\ll$ then $sos_\xi \dot{\equiv}_\theta sos'$.*

Proof ***(i):*** Follows from the fact that if $\Phi_j \cap \Delta_i \neq \varnothing$ then $\Delta_i \subseteq \Phi_j$ as Δ_i is an equivalence class of $\sqsubset^{\circledast}$.

(ii): Follows from Proposition 14.

(iii): By part (i), $\Phi_j = \Delta_{i_1} \uplus \dots \uplus \Delta_{i_l}$. Suppose that $l > 1$. Since $\ll$ is acyclic, there is $s \leq l$ such that there is no $z \in Z = \{i_1, \dots, i_l\} \setminus \{i_s\}$ with $\Delta_z \ll \Delta_{i_s}$ (i.e., Δ_{i_s} is $\ll$-minimal).

Consider next the nonempty sets Δ_{i_s} and $\Phi_j \setminus \Delta_{i_s}$. Suppose $\alpha \in \Delta_{i_s}$ and $\beta \in \Delta_z$, for some $z \in Z$, which means that $pos_u(\alpha) \neq pos_u(\beta)$. By Eq.(8) and $\alpha \not\rightleftharpoons_{sos'} \beta$, we have $\langle \ell(\alpha), \ell(\beta) \rangle \in \mathsf{sim}$. Suppose that $\langle \ell(\alpha), \ell(\beta) \rangle \notin \mathsf{seq}$. Then, by Eq.(8), $\beta \sqsubset \alpha$, contradicting the choice of Δ_{i_s} ($\ll$-minimality). As a result, $A_{i_s} \times \bigcup_{z \in Z} A_z \subseteq \mathsf{seq}$. Hence

$$sos'' = \mathsf{sseq2sos}(\ell(\Phi_1 \dots \Phi_{i_s - 1} \Delta_{i_s} (\bigcup_{z \in Z} \Delta_z) \Phi_{i_s + 1} \dots \Phi_m))$$

is such that $sos' \dot{\approx} sos''$ and $sos'' \in \mathsf{os2soss}(os)$. This produces a contradiction with the choice of sos'. Hence $\tau_{sos'}$ is a linearisation of $\ll$.

(iv): Let $\alpha \in X$ and $\beta \in Y$. Then, by Proposition 16(*ii*) (1st or 3rd line), we have $\langle \ell(\alpha), \ell(\beta) \rangle \in (\mathsf{sim} \cap \mathsf{seq} \cap \mathsf{seq}^{-1}) \cup (\mathsf{seq} \cap \mathsf{seq}^{-1} \setminus \mathsf{sim}) = \mathsf{seq} \cap \mathsf{seq}^{-1}$.

(v): By (*iii*), we can assume that $\tau_{sos'}$ is a linearisation of $\ll$. By Lemma 2, there are linearisations $v_1, \dots, v_r$ of $\ll$ such that $\tau_{sos'} = v_1 \sim \dots \sim v_k = \xi$. We then observe that the result follows from (*iv*). □

Lemma 8 *Let* $u \in \mathsf{SSEQ}^{thin}_\theta$ *and* $os = \mathsf{sseq2os}_\theta(u) = \langle \Delta, \rightleftharpoons, \sqsubset, \ell\rangle$. *Then* $\mathsf{sseq2sos}(u) \dot{\equiv}_\theta sos$, *for every* $sos \in \mathsf{os2soss}(os)$.

Proof Follows from Lemmata 3 and 7. □

We now observe that Proposition 20 follows Propositions 17(1) and 19, Lemma 8.

Proof of Theorem 7. Follows from Theorems 4 and 6, Proposition 18, and the following argument.

Let $u \in \mathsf{SSEQ}_\theta$. Suppose that $w \equiv_\theta u$. Then, by Proposition 17(1), $\mathsf{sseq2os}_\theta(w) = \mathsf{sseq2os}_\theta(u)$. Hence, by Proposition 17(1), $w \in \mathsf{sos2sseq} \circ \mathsf{os2soss} \circ \mathsf{sseq2os}_\theta(u)$. As a result, $[\![u]\!]_\theta \subseteq \mathsf{sos2sseq} \circ \mathsf{os2soss} \circ \mathsf{sseq2os}_\theta(u)$. Moreover, by Proposition 20, $\mathsf{sos2sseq} \circ \mathsf{os2soss} \circ \mathsf{sseq2os}_\theta(u) \subseteq [\![u]\!]_\theta$. Hence $[\![u]\!]_\theta = \mathsf{sos2sseq} \circ \mathsf{os2soss} \circ \mathsf{sseq2os}_\theta(u)$.

Proof of Proposition 21. Let $os = \langle \Delta, \rightleftharpoons, \sqsubset, \ell\rangle$. Since ℓ is injective, we may assume that each $\alpha \in \Delta$ is of the form $\langle a, 1\rangle$ with $\ell(\alpha) = a$. Hence Δ is an event domain.

By Proposition 11, there is $sos \in \mathsf{os2soss}(os) \neq \varnothing$. Clearly, $sos \in \mathsf{SOS}_\theta$, for any generalised concurrency alphabet $\theta = \langle \ell(\Delta), \mathsf{sim}, \mathsf{seq}\rangle$. We will now show how to construct sim and seq in order to obtain a desired alphabet.

First, we observe that the layer sequence τ_{sos} is well-defined even though θ is not fully defined. Moreover, τ_{sos} can be treated as a step sequence over the alphabet Δ. We then construct sim and seq as follows, by taking all pairs of distinct $\alpha, \beta \in \Delta$ with $k = pos_{\tau_{sos}}(\alpha)$ and $m = pos_{\tau_{sos}}(\beta)$:

Case 1: $\alpha \not\sqsubset \beta \wedge \beta \not\sqsubset \alpha \wedge \alpha \not\rightleftharpoons \beta$. Then we add $\langle \ell(\alpha), \ell(\beta)\rangle$ and $\langle \ell(\beta), \ell(\alpha)\rangle$ to both sim and seq.

Case 2: $\alpha \sqsubset \beta \wedge \beta \sqsubset \alpha \wedge \alpha \not\rightleftharpoons \beta$. Then $k = m$ and we add $\langle \ell(\alpha), \ell(\beta)\rangle$ and $\langle \ell(\beta), \ell(\alpha)\rangle$ to sim.

Case 3: $\alpha \not\sqsubset \beta \wedge \beta \not\sqsubset \alpha \wedge \alpha \rightleftharpoons \beta$. Then $k \neq m$ and we add $\langle \ell(\alpha), \ell(\beta)\rangle$ and $\langle \ell(\beta), \ell(\alpha)\rangle$ to seq.

Case 4: $\alpha \sqsubset \beta \wedge \beta \not\sqsubset \alpha \wedge \alpha \not\rightleftharpoons \beta$. Then $k \leq m$ and we add $\langle \ell(\alpha), \ell(\beta)\rangle$ and $\langle \ell(\beta), \ell(\alpha)\rangle$ to sim, and $\langle \ell(\alpha), \ell(\beta)\rangle$ to seq.

Case 5: $\alpha \sqsubset \beta \wedge \beta \not\sqsubset \alpha \wedge \alpha \rightleftharpoons \beta$. Then $k < m$ and we do not add anything.

Note that the above construction follows from the characterisation provided by Proposition 16(2).

We observe that θ is a generalised concurrency alphabet. Indeed, sim and seq are irreflexive by construction and the fact that $\alpha \neq \beta$ implies $\ell(\alpha) \neq \ell(\beta)$. Moreover, sim is symmetric by construction, and $\mathsf{seq} \setminus \mathsf{sim}$ is symmetric because it can only acquire pairs of elements in Case 3.

Let $u = \mathsf{sos2sseq}(sos) = \ell(\tau_{sos})$. Then $u \in \mathsf{SSEQ}_\theta$ follows from the fact that, in the above construction, if $k = m$ then we have Case 1 or Case 2 or Case 4. We then observe that $os = \mathsf{sseq2os}_\theta(u)$ follows from Eq.(8).