Structured Concurrency Semantics for Series-Parallel Orders

Maciej Koutny[1], Łukasz Mikulski[2,*]

[1] School of Computing, Newcastle University, Urban Sciences Building, 1 Science Square, Newcastle Helix, Newcastle upon Tyne, NE4 5TG, United Kingdom

[2] Faculty of Mathematics and Computer Science & Institute of Advanced Studies, Nicolaus Copernicus University in Toruń, Chopina 12/18, Toruń, Poland

Abstract

In formal semantics of concurrent systems, executions are typically represented as partial orders over events, capturing different aspects of sequentiality and concurrency, such as total, stratified, or interval orders. An alternative, more compositional perspective focuses on series-parallel orders, in which executions are constructed using sequential and parallel composition of sets of events, providing a natural way to reflect the compositional structure of concurrent behaviours. Moreover, since analysing all individual executions is often infeasible, a more effective approach is to reason at a higher level of abstraction using relational specifications that capture invariant relationships between events, such as causality or temporal constraints.

In this paper, we introduce a new model of such specifications for series-parallel executions, based on two relations: precedence ("earlier than") and weak precedence ("not later than"). These relations provide a compact way to describe sets of executions and capture key concurrency phenomena such as causality and simultaneity. We study the structural properties of these specifications and establish their correspondence with series-parallel executions, yielding a new order-theoretic framework for compositional reasoning about concurrency.

Keywords

partial order, total order, stratified order, interval order, composition, series-parallel interval order, concurrency, precedence and weak precedence, causality, simultaneity, relational structure, closure

1. Introduction

This paper investigates how partial orders can be employed to represent executions of concurrent systems, where each execution consists of a collection of events, understood as occurrences of actions or operations. A number of semantic models of concurrency rely on particular classes of partial orders to describe the ways in which events are arranged along an execution. In such representations it is always possible to determine the relative relationship between any pair of events, specifying whether one precedes the other or whether the two occur simultaneously. Consequently, these structures can naturally be interpreted as abstract observations of discrete physical processes.

Throughout the paper we interpret events (that is, executed actions) as activities occurring over time intervals. We do not aim to capture the exact temporal duration of such intervals. Rather, we are concerned only with their qualitative temporal relationships. In particular, we distinguish whether two events are strictly ordered (meaning that the interval corresponding to one event precedes the interval corresponding to the other) or whether they are simultaneous (in the sense that their corresponding intervals overlap).

In the literature, one can identify at least three classes of partial orders that have been used to model concurrent executions according to the intuition outlined above. These include (i) total orders, representing purely sequential executions; (ii) stratified orders, representing behaviours in which sets of simultaneous events are executed sequentially as steps; and (iii) interval orders, representing behaviours where events may occupy time intervals and where overlapping intervals correspond to simultaneity

International Workshop on Petri Nets, Higher-Dimensional Automata, Partial Orders, and Concurrency, June, 2026, Hamburg, Germany

*Corresponding author.

maciej.koutny@ncl.ac.uk (M. Koutny); lukasz.mikulski@mat.umk.pl (Ł. Mikulski)

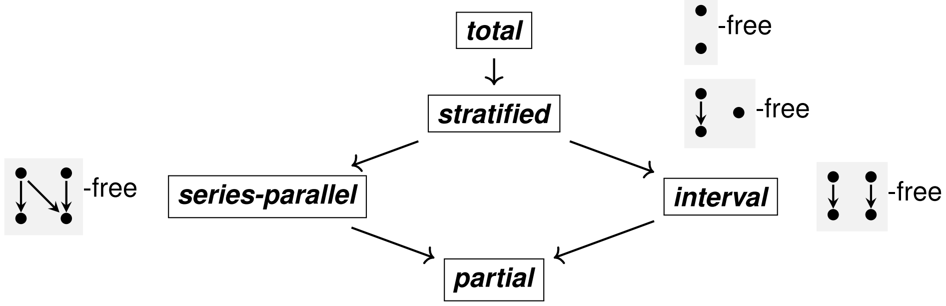

Figure 1: A strict inclusion hierarchy (indicated by directed arcs) of partial order classes considered in this paper. Except for the general partial orders, each class is characterised by 'forbidden' pattern(s).

of events. Representations of the latter kind were regarded by Wiener [1] as the most general form of physical observation available to a single observer.

Each of the above classes admits a clear operational interpretation. From the perspective of modelling concurrent behaviour, total and stratified orders provide well-structured representations that reflect natural execution patterns. Interval orders, by contrast, capture more general temporal observations but do not intrinsically encode structural information about the executions they represent. One of the principal aims of this paper is therefore to identify and investigate a non-trivial class of partial orders that admits meaningful structural characterisation.

A particularly important class of partial orders endowed with well-defined structural properties are the series-parallel orders (SP-orders). These structures are defined using two composition operators (sequential and parallel composition) which correspond to fundamental mechanisms used in the design and analysis of concurrent systems. Importantly, these operators apply to sets of events rather than to individual events, thereby enabling structured descriptions of concurrent behaviour. Despite the central role played by SP-orders in concurrency theory, the structural properties of their prototype specifications and of the sets of executions implementing them have not yet been systematically investigated.

Figure 1 depicts the (strict) hierarchy of the classes of partial orders considered in this paper. An interesting observation is that every class in this hierarchy, except for the most general one, admits a characterisation in terms of forbidden patterns (subgraphs). Such forbidden configurations provide useful insights into the structural properties of the corresponding classes of executions.

The properties of SP-orders, and in particular their algebraic and order-theoretic aspects, have been studied extensively in the literature; see, for example, [2, 3]. The algebraic framework developed in [2] enabled the work of [4] to provide a partial-order semantic treatment for a simple process algebra equipped with sequential and parallel composition operators. As another example, [5, 6] extended classical automata theory from languages of words to languages consisting of labelled SP-orders. The relationship between SP-orders and the K-density property in Petri nets was investigated in [7]. A complete axiomatisation of inclusion for SP-orders was obtained in [8]. More recently, the use of series-parallel orders as a foundation for automata and logical formalisms has been explored in [9], while [10] proposed a generalisation of SP-orders allowing more flexible composition operators.

Directly reasoning about the potentially vast number of executions of a concurrent system is generally infeasible. A more effective approach is to consider executions at a higher level of abstraction, using behavioural specifications that capture relationships between events, such as causality or dependency. Such a specification is typically represented by a relational structure *rs*, which characterises a (possibly large) set *RS* of individual executions. Examples of such relational structures can be found, for instance, in [11, 12, 13, 14]. These include causal partial orders for sequences and invariant structures for step sequences. Structures of this kind provide concise representations of concurrent histories and their usefulness follow from generalisations of Szpilrajn's Theorem [15], which states that every partial order can be expressed as the intersection of its total-order extensions. Although the relational structure *rs* provides a clean theoretical characterisation of the set *RS*, turning such representations into practical tools (as, for example, in [16]) requires that they can be derived directly from individual executions using structural properties of the concurrent system.

This observation naturally leads to relational representations based on acyclic relations between events, such as the dependence graphs introduced in [17]. Applying transitive closure to such acyclic relations yields invariant structures capturing the essential causal structure of executions.

The above perspective has been proposed and investigated in [11, 12] as a general methodological framework for constructing semantic models based on relational structures. In the present paper, we apply this approach to the study of specifications of executions given by SP-orders.

1.1. About this paper

This paper lays the foundations for a novel concurrency semantics based on SP-orders and introduces their specifications in the form of *series-parallel relational structures*. The proposed framework follows the general methodology for defining order-theoretic semantics of concurrent behaviours developed in [11, 12]. Within this approach, sets of executions are characterised through relational structures capturing invariant relationships between events.

The specifications we propose rely on two fundamental relationships between events, namely *precedence* and *weak precedence*. Intuitively, these relations correspond to the relative temporal positions of events in individual executions: precedence expresses the fact that one event occurs strictly earlier than another, whereas weak precedence captures the more permissive situation in which one event is *not later than* another. Together, these relations provide a natural and expressive mechanism for capturing causal and temporal constraints that arise in concurrent computations.

The same pair of relations has been studied previously in [12, 18, 13], where relational structures based on them were shown model a variety of computational frameworks, including inhibitor nets, Boolean networks, reaction systems, and membrane systems [12, 19]. Building on these results, the present work adapts this relational perspective to the setting of series-parallel executions. The resulting SP-order–based specifications provide a structured representation of concurrent behaviours while retaining the generality needed to capture rich classes of executions. In particular, the proposed model is strictly more expressive than the *Set-Sequential Object* model and incomparable with the *Interval-Sequential Object* model in the hierarchy presented in Fig. 1 of [20].

1.2. Contributions

The main contributions of this paper can be summarised as follows:

- We introduce a novel semantic framework for concurrent executions based on SP-orders, providing a structured representation of behaviours that naturally combines sequential and parallel composition patterns.
- We propose *series-parallel relational structures* as specifications of sets of SP-orders, following the invariant-structure methodology developed in [11, 12]. These structures capture executions through two fundamental relations between events: precedence and weak precedence.
- We investigate the structural properties of these specifications and establish their correspondence with sets of executions represented by SP-orders, thereby providing a systematic framework for reasoning about series-parallel concurrent behaviours.

1.3. Organization of the paper

The next section recalls basic notions and terminology concerning partial orders. Section 3 reviews relational structures used to specify sets of executions. Section 4 introduces *series-parallel relational structures*. Section 5 characterises maximal SP-structures and relates them to SP-orders. Section 6 studies closed SP-structures and the associated closure operator. The final section provides concluding remarks.

2. Partial orders modelling concurrent behaviours

In this paper, observed behaviours (executions) of concurrent systems are represented by partial orders capturing event precedence (the lack of ordering is interpreted as simultaneity).

Let Δ be a finite domain (set of events) and $\ll$ be an irreflexive binary (precedence) relation over Δ. Moreover, let $\frown$ be a binary (simultaneity) relation over Δ comprising all pairs $\langle x,y\rangle$ such that $x \neq y$ and $x \not\ll y \not\ll x$. Then $po = \langle \Delta, \ll\rangle$ is called (below $x,y,z,w \in \Delta$):

- (strict) partial order if $\boxed{x \ll z \ll y \implies x \ll y}$
- total order if $\boxed{(x \neq y \wedge y \not\ll x) \vee x \ll z \ll y \implies x \ll y}$
- stratified order if $\boxed{x \ll z \ll y \vee x \frown z \ll y \vee x \ll z \frown y \implies x \ll y}$
- interval order if $\boxed{x \ll y \wedge z \ll w \implies x \ll w \vee z \ll y}$
- series-parallel order (SP-order) if $\boxed{x \ll z \ll y \implies x \ll y}$ and $\boxed{z \ll y \frown w \wedge z \frown x \ll w \wedge z \ll w \implies x \ll y}$

The sets of partial / total / stratified / interval / SP- orders are denoted by *PO* / *TO* / *SO* / *IO* / *SPO*. A trivial partial order $po = \langle \Delta, \varnothing\rangle$ will be denoted by π_Δ.

Figure 1 depicts inclusions between different classes of partial orders. Partial orders belonging to each class $\mathscr{X} \in \{TO, SO, IO, SPO\}$, can be characterised by the absence of forbidden pattern(s) depicted in Figure 1. Hence each class is projection-closed, i.e., if $po = \langle \Delta, \ll\rangle \in \mathscr{X}$ and $\Phi \subseteq \Delta$, then $po|_\Phi = \langle \Phi, \ll \cap (\Phi \times \Phi)\rangle \in \mathscr{X}$.

Series-parallel orders can be defined compositionally (which is reflected in their name). One needs two ways of composing partial orders with disjoint domains, $po = \langle \Delta, \ll\rangle$ and $po' = \langle \Delta', \ll'\rangle$:

- $spo \otimes spo' = \langle \Delta \cup \Delta', (\ll \cup \ll') \cup (\Delta \times \Delta')\rangle$. (series composition)
- $spo \oplus spo' = \langle \Delta \cup \Delta', \ll \cup \ll'\rangle$. (parallel composition)

Both operators are associative and have $\pi_\varnothing$ as the unit; moreover, $\oplus$ is commutative. Then SP-orders can be shown to be the smallest set *SPO* comprising all trivial partial orders and such that, for all $spo, spo' \in SPO$ with disjoint domains, $spo \oplus spo'$ and $spo \otimes spo' \in SPO$. Following this, one can use algebraic expressions like $\pi_{\{x\}} \otimes (\pi_{\{y\}} \oplus \pi_{\{z\}})$ to denote SP-orders.

3. Relational structures for specifying system executions

There is a distinction between specifications of concurrent system behaviour (e.g., given as sets of constraints about the ordering of actions to be executed), and the actual observed executions. In particular, a single specification can be realised by (very) many executions.

In this paper, we consider structures with two relationships 'generating' executions consistent with these two relations. Such a *structure* is a triple $rs = \langle \Delta, \prec, \sqsubset\rangle$, where $\prec$ and $\sqsubset$ are two binary relations over a finite domain Δ. Moreover, if both relations are irreflexive, then rs is a *relational structure*. Intuitively, $x \prec y$ (precedence) is a record that event x preceded event y, whereas $x \sqsubset y$ (weak precedence) is a record that x preceded or was simultaneous with y (i.e., y did not precede x)[1]. Also, if $x \sqsubset y \sqsubset x$ then x and y are considered to be simultaneous. We use Δ_{rs}, $\prec_{rs}$, and $\sqsubset_{rs}$ to denote the components of rs, and the notations below can be used to enlarge these components, where $x \notin \Delta$, $y,z \in \Delta$ and $y \neq z$:

- $rs[x] = \langle \Delta \cup \{x\}, \prec, \sqsubset\rangle$
- $rs[y \longrightarrow z] = \langle \Delta, \prec \cup \{\langle y,z\rangle\}, \sqsubset\rangle$
- $rs[y \dashrightarrow z] = \langle \Delta, \prec, \sqsubset \cup \{\langle y,z\rangle\}\rangle$.

Let $\mathscr{R}$ be a set of relational structures, rs, rs' be relational structures, and $\Phi \subseteq \Delta_{rs}$. Then:

[1] In intended execution models, x precedes y ($x \prec y$) excludes y weakly precedes x ($y \sqsubset x$). In maximal structures, $\sqsubset$ can therefore be seen as the complement of the reversed $\prec$.

- $rs \trianglelefteq rs'$ means that rs' is an *extension* of rs, i.e., $\Delta_{rs} = \Delta_{rs'}$, $\prec_{rs} \subseteq \prec_{rs'}$, and $\sqsubset_{rs} \subseteq \sqsubset_{rs'}$.
- $rs \triangleleft rs'$ means that rs' is a *strict extension* of rs i.e., $rs \trianglelefteq rs'$ and $rs \neq rs'$.
- $\text{ext}_{\mathscr{R}}(rs) = \{rs' \in \mathscr{R} \mid rs \trianglelefteq rs'\}$ are the *extensions of rs in $\mathscr{R}$.*
- $\mathscr{R}^{max} = \{rs' \in \mathscr{R} \mid \text{ext}_{\mathscr{R}}(rs') = \{rs'\}\}$ are the *maximal structures* in $\mathscr{R}$.
- $\max_{\mathscr{R}}(rs) = \text{ext}_{\mathscr{R}^{max}}(rs)$ are the *maximal extensions* (or saturations) of rs in $\mathscr{R}$.
- $rs|_\Phi = \langle \Phi, \prec|_{\Phi\times\Phi}, \sqsubset|_{\Phi\times\Phi}\rangle$ is the projection of rs onto Φ.
- $\mathscr{R}$ is *projection-closed* if $rs|_\Phi = \langle \Phi, \prec|_{\Phi\times\Phi}, \sqsubset|_{\Phi\times\Phi}\rangle \in \mathscr{R}$, for all $rs \in \mathscr{R}$ and $\Phi \subseteq \Delta_{rs}$.
- $\mathscr{R}$ is *left-closed* if $rs' \in \mathscr{R}$, for every $rs \in \mathscr{R}$ and every structure rs' satisfying $rs' \trianglelefteq rs$.
- $\mathscr{R}$ is *intersection-closed* if $\langle \Delta, \prec_{rs} \cap \prec_{rs'}, \sqsubset_{rs} \cap \sqsubset_{rs'}\rangle \in \mathscr{R}$ whenever $rs, rs' \in \mathscr{R}$ and $\Delta = \Delta_{rs} = \Delta_{rs'}$.
- $G_{rs} = \langle \Delta_{rs}, \prec_{rs} \cup \sqsubset_{rs}\rangle$ is a directed graph jointly representing both relations of rs.

A partial order $po = \langle \Delta, \ll\rangle$ can be represented as relational structure $\rho(po) = \langle \Delta, \ll, \sqsubset\rangle$, where $\sqsubset = \prec \cup \frown$. Hence, $x \frown y$ in po is represented by $x \sqsubset y \sqsubset x$ in $\rho(po)$.

Relational structures in $\mathscr{R}^{max}$ are intended to represent executions specified by relational structures in $\mathscr{R}$. Suppose then that $\mathscr{X}$ is a set of partial orders representing concurrent executions and that we would like to characterise the set of *all* relational structures $\mathscr{R}$ which are *non-vacuous specifications for* $\mathscr{X}$, i.e., all relational structures rs satisfying $\max_{\mathscr{R}}(rs) \cap \rho(\mathscr{X}) \neq \varnothing$ (that is, each partial order $po \in \mathscr{X}$ with $\rho(po) \in \max_{\mathscr{R}}(rs)$ is regarded an *execution* of rs). One can see that such a set $\mathscr{R}$ is characterised by being left-closed and satisfying $\mathscr{R}^{max} = \rho(\mathscr{X})$. In such a case, the set of partial order executions specified by $rs \in \mathscr{R}$ is given by $\rho^{-1}(\max_{\mathscr{R}}(rs))$.

For some of the partial order classes modelling executions in Section 2, all non-vacuous specifications $rs = \langle \Delta, \prec, \sqsubset\rangle$ can be characterised in the following way (see [12, 18, 21]):

- There is a total order execution of rs iff there are no cycles in G_{rs}.
- There is a stratified order execution of rs iff there is no cycle $x_1, \dots, x_n$ in G_{rs} such that $x_1 \prec x_2$.
- There is an interval order execution of rs iff there is no cycle $x_1, \dots, x_n$ in G_{rs} such that:
  (i) $x_{i-1} \prec x_i \ \vee \ x_i \prec x_{i+1}$, for all $1 < i \leq n$; and (ii) $x_{n-1} \prec x_n \ \vee \ x_n \prec x_1$.

Each case above captures a kind of 'acyclicity' or 'constrained cyclicity'. Such a property is linked to the absence of substructures of rs which are 'forbidden' in a given execution model. For example, if $x \sqsubset y \sqsubset x$ then no total order exists in which x does not precede y and, at the same time, y does not precede x. However, in the case of SP-orders the 'constrained cyclicity' conditions are no longer sufficient, and we will propose to use instead a 'constrained strong connectivity' condition.

4. Series-parallel structures

To introduce relational structures for specifying concurrent executions modelled by SP-orders we need auxiliary notions.

Let $rs = \langle \Delta, \prec, \sqsubset\rangle$ be a structure and $\Phi \subseteq \Delta$ be such that $|\Phi| \geq 2$. Then:

- A partition $\Xi \uplus \Psi$ of Φ is an *rs-partition* if $(\Xi \times \Psi \cup \Psi \times \Xi) \cap \prec = \varnothing$.
- Φ is a *strongly connected set* in rs if it induces a strongly connected sub-graph of G_{rs}. We denote this by $\Phi \in SCS(rs)$.
- If $\Phi \in SCS(rs)$ and there is no rs-partition of Φ, then Φ is an *indivisible strongly connected set (or* ISC-*set) in rs* . We denote this by $ISCS(rs)$.

Proposition 1. *Let rs and rs′ be structures such that $rs \trianglelefteq rs'$, and $\Phi \subseteq \Delta_{rs}$. Then:*

1. *$SCS(rs) \subseteq SCS(rs')$ and $ISCS(rs) \subseteq ISCS(rs')$.*
2. *$SCS(rs|_\Phi) \subseteq SCS(rs)$ and $ISCS(rs|_\Phi) \subseteq ISCS(rs)$.*

Proof. (1) It follows directly from the definitions, and observing that: (i) $\prec_{rs} \subseteq \prec_{rs'}$, and (ii) adding additional arcs maintains the property of being a strongly connected sub-graph.

(2) It follows from the fact the all relationships within Φ in $rs|_\Phi$ are the same as in rs. ☐

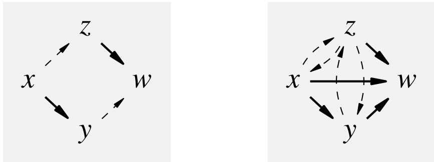

Figure 2: A SP-structure together with one of its SPM-structure extensions. Solid and dashed arcs represent relations $\prec \subseteq \sqsubset$ and $\sqsubset \setminus \prec$, respectively.

Proposition 1 shows the properties of two operations that will be used repeatedly later: extending a structure and restricting it to a smaller domain. This is exactly what one wants from forbidden patterns, viz. any problematic fragment of a structure should be detectable locally. It is therefore natural to take an absence of indivisible strongly connected subsets as the defining property for series-parallel specifications.

Definition 1 (SP-structure)**.** *A structure rs is* SP-acyclic *if $ISCS(rs) = \varnothing$. Then, a* series-parallel structure *(or* SP*-structure) is a relational structure which is* SP*-acyclic. The set of all* SP*-structures is denoted by SPS.*

Proposition 2. SP*-structures are projection-closed, left-closed, and intersection-closed.*

Proof. The first two properties follow directly from Proposition 1. Moreover, the third one follows directly from the second one. □

In the context of the discussion on 'constrained strong cyclicity/connectivity' at the end of the last section, the role of 'forbidden cyclic substructures' is played in SP-structures by the indivisible strongly connected subsets.

5. Maximal SP-structures

We now provide an axiomatic characterisation of maximal SP-structures.

Definition 2 (SPM-structure)**.** *A* series-parallel maximal structure *(or* SPM*-structure) is a structure $spms = \langle \Delta, \prec, \sqsubset \rangle$ such that the following hold, for all $x, y, z, t \in \Delta$:*

- SPMS:1 $x \not\sqsubset x$
- SPMS:2 $x \prec y \iff x \sqsubset y \not\sqsubset x$
- SPMS:3 $x \neq y \iff x \prec y \;\vee\; y \prec x \;\vee\; x \sqsubset y \sqsubset x$
- SPMS:4 $x \prec y \prec z \implies x \prec z$
- SPMS:5 $x \sqsubset z \sqsubset x \;\wedge\; y \sqsubset w \sqsubset y \;\wedge\; x \prec y \;\wedge\; z \prec w \;\wedge\; z \prec y \implies x \prec w.$

The set of all SPM*-structures is denoted by SPMS.*

Clearly, SPM-structures are projection-closed.

Figure 2 shows a SP-structure together with one possible SPM-structure extending it. The SPM-structure on the right corresponds (through the mapping ρ introduced in Section 3) to the SP-order $((\pi_{\{x\}} \otimes \pi_{\{y\}}) \oplus \pi_{\{z\}}) \otimes \pi_{\{w\}}$.

The next result clarifies relationship between SP-orders and SPM-structures. Basically, it demonstrates that $\rho(SPO) = SPMS$ (see Section 3).

Proposition 3. *A structure $sps = \langle \Delta, \prec, \sqsubset \rangle$ belongs to SPMS iff the following two properties are satisfied: (i) $\langle \Delta, \prec \rangle \in SPO$, and (ii) $\sqsubset = \{\langle x, y \rangle \in \Delta \times \Delta \mid x \neq y \;\wedge\; y \not\prec x\}$.*

Proof. Let $po = \langle \Delta, \prec \rangle$ and $S = \{\langle x,y\rangle \in \Delta \times \Delta \mid x \neq y \ \wedge \ y \not\prec x\}$.

($\Longrightarrow$) By SPMS:1,2, $\prec$ is irreflexive. This and SPMS:4 yields $po \in PO$. Hence, by SPMS:5 and the definition of SP-orders, $po \in SPO$ and so (i) holds.

To show $S \subseteq \sqsubset$, suppose that $x \neq y \in \Delta$ and $y \not\prec x$. Then, by SPMS:3, $x \sqsubset y \sqsubset x$ or $x \prec y$. Moreover, in the latter case, we have $x \sqsubset y$ by SPMS:2. Hence, the $S \subseteq \sqsubset$ inclusion holds.

To show the $\sqsubset \subseteq S$ inclusion, suppose that $x \sqsubset y$. Then, by SPMS:1, $x \neq y$. If $y \prec x$ then, by SPMS:2, $x \not\sqsubset y$, a contradiction. Hence $y \not\prec x \neq y$ and so the $\sqsubset \subseteq S$ inclusion holds.

($\Longleftarrow$) We consider the following cases.

- (SPMS:1) It follows from (ii).
- (SPMS:2, $\Longrightarrow$) Suppose $x \prec y$. Then, by $po \in PO$, $y \not\prec x \neq y$. Hence, by (ii), $x \sqsubset y \not\sqsubset x$.
- (SPMS:2, $\Longleftarrow$) Suppose $x \sqsubset y \not\sqsubset x$. Then, by SPMS:1 (see above), $x \neq y$. Hence, by (ii), $x \prec y$.
- (SPMS:3, $\Longrightarrow$) Suppose $x \neq y$ and $x \not\prec y \not\prec x$. Then, by (ii), $x \sqsubset y$ and $y \sqsubset x$.
- (SPMS:3, $\Longleftarrow$) It follows from SPMS:1,2 (see above).
- (SPMS:4) It follows from $po \in PO$.
- (SPMS:5) Suppose that $x \sqsubset z \sqsubset x \ \wedge \ y \sqsubset w \sqsubset y \ \wedge \ x \prec y \ \wedge \ z \prec w \ \wedge \ z \prec y$. Then, by (ii) and $x \sqsubset z \sqsubset x \ \wedge \ y \sqsubset w \sqsubset y$, we have $z \frown_{po} x \ \wedge \ y \frown_{po} w$. Therefore, since $po \in SPO$, we obtain $x \prec w$ by the definition of SP-orders.

$\square$

Proposition 3 is a key representation result for maximal structures. It shows that elements of *SPMS* are not an alternative semantics competing with SP-orders, but exactly their relational encodings. In particular, once the precedence relation is fixed in the maximal case, the weak precedence relation is determined canonically.

We are now in a position to relate the maximal SP-structures with their relational encodings as well as axiomatically defined SPM-structures.

Theorem 1. $\rho(SPO) = SPMS = SPS^{max}$.

Proof. The first equality follows from Proposition 3. The second one is proved below.

($SPMS \subseteq SPS^{max}$) Let $spms = \langle \Delta, \prec, \sqsubset \rangle \in SPMS$ and $spo = \langle \Delta, \prec \rangle$. By Proposition 3, $spo \in SPO$. We first show that $spms \in SPS$, proceeding by induction on structure of spo.

If $spo = \pi_\varnothing$ then, clearly, $spms \in SPS$. In the inductive step, we consider two cases.

Case 1: $spo = spo' \oplus spo''$, where $spo', spo'' \in SPO$ and $\Delta_{spo'} \neq \varnothing \neq \Delta_{spo''}$. Then, by the induction hypothesis and the fact that SPM-structures are projection-closed (see Proposition 2),

$$spms' = spms|_{\Delta_{spo'}} \in SPS \ \text{ and } \ spms'' = spms|_{\Delta_{spo''}} \in SPS \,.$$

Let $\Phi \in SCS(spms)$. If $\Phi \subseteq \Delta_{spo'}$ or $\Phi \subseteq \Delta_{spo''}$ then, by $spms', spms'' \in SPS$, we have $\Phi \notin ISCS(spms') \cup ISCS(spms'')$. Hence $\Phi \notin ISCS(spms)$. If

$$\Phi' = \Phi \cap \Delta_{spo'} \neq \varnothing \neq \Phi'' = \Phi \cap \Delta_{spo''} \,,$$

then $\Phi' \uplus \Phi''$ is an *spms*-partition of Φ. Hence, $\Phi \notin ISCS(spms)$. As a result, we obtained that $spms \in SPS$.

Case 2: $spo = spo' \otimes spo''$, where $\Delta_{spo'} \neq \varnothing \neq \Delta_{spo''}$. Then, by the induction hypothesis and the fact that SPM-structures are projection-closed, we have

$$spms' = spms|_{\Delta_{spo'}} \in SPS \ \text{ and } \ spms'' = spms|_{\Delta_{spo''}} \in SPS \,.$$

Moreover, $(\prec \cup \sqsubset) \cap (\Delta_{spo''} \times \Delta_{spo'}) = \varnothing$, which means that $SCS(spms) = SCS(spms') \cup SCS(spms'')$, and so $ISCS(spms) = \varnothing$. As a result, we obtained that $spms \in SPS$.

Hence $spms \in SPS$. Moreover, by Proposition 3, for all $x \neq y \in \Delta$, $x \not\prec y$ implies $\{x,y\} \in ISCS(spms[x \longrightarrow y])$ and $x \not\sqsubset y$ implies $\{x,y\} \in ISCS(spms[x \dashrightarrow y])$. Hence $spms \in SPS^{max}$.

($SPS^{max} \subseteq SPMS$) Since SP-structures are finite and we have already proved that $SPMS \subseteq SPS$, it suffices to show that $\mathrm{ext}_{SPMS}(sps) \neq \varnothing$, for every $sps = \langle \Delta, \prec, \sqsubset \rangle \in SPS$.

We proceed by induction on the size of the domain of sps. Clearly, $\langle \varnothing, \varnothing, \varnothing \rangle \in \mathrm{ext}_{SPMS}(\langle \varnothing, \varnothing, \varnothing \rangle)$. In the inductive step $|\Delta| > 1$ and we consider two cases.

Case 1: $\Delta \in SCS(sps)$. Then, by $sps \in SPS$, Δ can be partitioned as $\Phi' \uplus \Phi''$ in such a way that $((\Phi' \times \Phi'') \cup (\Phi'' \times \Phi')) \cap \prec = \varnothing$. As SP-structures are projection-closed, $sps|_{\Phi'}, sps|_{\Phi''} \in SPS$. Hence, by the induction hypothesis, there are $spms' \in \mathrm{ext}_{SPMS}(sps|_{\Phi'})$ and $spms'' \in \mathrm{ext}_{SPMS}(sps|_{\Phi''})$. A routine verification shows that $\langle \Delta, \prec_{spms'} \cup \prec_{spms''} \rangle \in SPO$ and so, by Proposition 3,

$$\langle \Delta, \prec_{spms'} \cup \prec_{spms''}, \sqsubset_{spms'} \cup \sqsubset_{spms''} \cup (\Phi' \times \Phi'') \cup (\Phi'' \times \Phi') \rangle$$

belongs to $\mathrm{ext}_{SPMS}(sps)$.

Case 2: $\Delta \notin SCS(sps)$. Then Δ can be partitioned as $\Phi' \uplus \Phi''$ so that $(\Phi'' \times \Phi') \cap (\prec \cup \sqsubset) = \varnothing$. As SP-structures are projection-closed, we have $sps|_{\Phi'} \in SPS$ and $sps|_{\Phi''} \in SPS$. Hence, by the induction hypothesis, there are $spms' \in \mathrm{ext}_{SPMS}(sps|_{\Phi'})$ and $spms'' \in \mathrm{ext}_{SPMS}(sps|_{\Phi''})$. It is easy to check that $\langle \Delta, \prec_{spms'} \cup \prec_{spms''} \cup (\Phi' \times \Phi'') \rangle \in SPO$ and so, by Proposition 3,

$$\langle \Delta, \prec_{spms'} \cup \prec_{spms''} \cup (\Phi' \times \Phi''), \sqsubset_{spms'} \cup \sqsubset_{spms''} \cup (\Phi' \times \Phi'') \rangle$$

belongs to $\mathrm{ext}_{SPMS}(sps)$. □

Theorem 1 together with Proposition 2 implies that SPS is the set of all non-vacuous specifications for SP-orders, as SPS turned out to be a left-closed set of relational structures satisfying $SPS^{max} = \rho(SPO)$. Moreover, by Theorem 1, having SPM-structure extensions yields an alternative characterisation of SP-structures.

Proposition 4. *A structure is an* SP*-structure iff one of its extensions is an* SPM*-structure.*

The next result captures some cases where extending (in the sense of the general relational structure extension) an SP-structure may (or may not) lead outside SP-structures. Intuitively, any valid extension of SP-structure is therefore SP-structure and repeated application of Proposition 5 together with the checks from Definition 1 can be used to saturate non-maximal SP-structures.

Proposition 5. *Let* $sps = \langle \Delta, \prec, \sqsubset \rangle \in SPS$ *and* $x \neq y \in \Delta$. *Then:*

1. $x \prec y$ *implies* $sps[y \dashrightarrow x] \notin SPS$.
2. $x \sqsubset y$ *implies* $sps[y \longrightarrow x] \notin SPS$.
3. $sps[x \longrightarrow y] \in SPS$ *or* $sps[y \dashrightarrow x] \in SPS$.
4. $sps[x \longrightarrow y] \notin SPS$ *implies* $sps[y \dashrightarrow x] \in SPS$ *and* $\max_{SPS}(sps[y \dashrightarrow x]) = \max_{SPS}(sps)$.
5. $sps[x \dashrightarrow y] \notin SPS$ *implies* $sps[y \longrightarrow x] \in SPS$ *and* $\max_{SPS}(sps[y \longrightarrow x]) = \max_{SPS}(sps)$.

Proof. (1) It follows from $\{x, y\} \in ISCS(sps[y \dashrightarrow x])$.

(2) It follows from $\{x, y\} \in ISCS(sps[y \longrightarrow x])$.

(3) Suppose that $rs_1 = sps[x \longrightarrow y] \notin SPS$ and $rs_2 = sps[y \dashrightarrow x] \notin SPS$.

As rs_i (for $i = 1, 2$) is a relational structure and $rs_i \notin SPS$, there is $\Phi_i \in ISCS(rs_i)$. From $sps \in SPS$ it follows that $x, y \in \Phi_1 \cap \Phi_2$. Moreover, $\Phi_1 \in ISCS(rs_1)$ implies that there is a directed path in G_{sps} from y to x going through the elements of Φ_1, and $\Phi_2 \in ISCS(rs_2)$ implies that there is a path in G_{sps} from x to y going through the elements of Φ_2. Hence $\Phi_1 \cup \Phi_2 \in SCS(sps)$, and so from $sps \in SPS$ it follows that there is an sps-partition $\Xi_1 \uplus \Xi_2$ of $\Phi_1 \cup \Phi_2$.

Let $\Psi_{ij} = \Phi_i \cap \Xi_j$ (for $i, j = 1, 2$). We consider two cases.

Case 1: $\Psi_{11} = \varnothing$ or $\Psi_{12} = \varnothing$. Then, since $\Phi_1 \cap \Phi_2 \neq \varnothing$, $\Psi_{21} \uplus \Psi_{22}$ is a partition of Φ_2. Moreover, since $y \sqsubset x$ is the only relationship added to get rs_2, $\Psi_{21} \uplus \Psi_{22}$ is an rs_2-partition of Φ_2, a contradiction.

Case 2: $\Psi_{11} \neq \varnothing$ and $\Psi_{12} \neq \varnothing$. Then $\Psi_{11} \uplus \Psi_{12}$ is a partition of Φ_1. Moreover, since $x \prec y$ is the only relationship added to get rs_1, we must have w.l.o.g. $x \in \Xi_1$ and $y \in \Xi_2$. Hence, $\Psi_{21} \uplus \Psi_{22}$ is a partition of Φ_2, and we can proceed as in Case 1 to obtain a contradiction.

(4) By part (3), $sps[y \dashrightarrow x] \in SPS$. Clearly, $\max_{SPS}(sps[y \dashrightarrow x]) \subseteq \max_{SPS}(sps)$. Suppose that $spms \in \max_{SPS}(sps) \setminus \max_{SPS}(sps[y \dashrightarrow x])$. Then we must have $y \not\sqsubset_{spms} x$, and so $x \prec_{spms} y$. As a result, $spms \in \max_{SPS}(sps[x \longrightarrow y])$. Hence, by Proposition 4, $sps[x \longrightarrow y] \in SPS$, a contradiction.

(5) Similarly as for part (4). □

6. Closed SP-structures

SP-structures are all the non-vacuous specifications for SP-order executions. We now identify SP-structures which are particularly suitable for, e.g., property verification or causality analysis in concurrent behaviours. According to the general theory expounded in [12] and the fact that SP-structures are intersection-closed (see Proposition 2), a class of such SP-structures can be defined as follows.

Definition 3. *An* SP-*structure* $sps \in SPS$ *is* closed *if*

- $sps = \langle \Delta_{sps}, \bigcap\{\prec_{spms} \mid spms \in \max_{SPS}(sps)\}, \bigcap\{\sqsubset_{spms} \mid spms \in \max_{SPS}(sps)\}\rangle$.

The set of all closed SP-*structures is denoted by* SPS^{clo}.

The above definition captures a generalisation of Szpilrajn's Theorem [15] stating that a partial order is the intersection of all its total order extensions.

From the general results proved in [12] (Prop.7.5), and the fact that SPS is a set of intersection-closed relational structures (see Proposition 2), it follows that there is a *structure closure* $\mathrm{clo}_{SPS} : SPS \to SPS^{clo}$ such that the following hold, for all $rs \in SPS^{clo}$ and $\overline{rs}, \widehat{rs} \in SPS$:

- $\boxed{\mathrm{clo}_{SPS}(\overline{rs}) = \langle \Delta_{\overline{rs}}, \bigcap\{\prec_{spms} \mid spms \in \max_{SPS}(\overline{rs})\}, \bigcap\{\sqsubset_{spms} \mid spms \in \max_{SPS}(\overline{rs})\}\rangle}$
- $\boxed{rs \lhd \overline{rs} \implies \max_{SPS}(\overline{rs}) \subset \max_{SPS}(rs)}$ extending rs excludes some SPM-structures
- $\boxed{\overline{rs} \unlhd \mathrm{clo}_{SPS}(\overline{rs}) = \mathrm{clo}_{SPS}(\mathrm{clo}_{SPS}(\overline{rs}))}$ clo_{SPS} is closure in the usual sense
- $\boxed{\max_{SPS}(\overline{rs}) = \max_{SPS}(\mathrm{clo}_{SPS}(\overline{rs}))}$ closure preserves generated SPM-structures
- $\boxed{\overline{rs} \unlhd \widehat{rs} \implies \mathrm{clo}_{SPS}(\overline{rs}) \unlhd \mathrm{clo}_{SPS}(\widehat{rs})}$ closure is monotonic
- $\boxed{\max_{SPS}(\overline{rs}) = \max_{SPS}(\widehat{rs}) \iff \mathrm{clo}_{SPS}(\overline{rs}) = \mathrm{clo}_{SPS}(\widehat{rs})}$ generating the same SPM-structures is equivalent to having the same closure.

The generic definition of closed SP-structures in Definition 3 is impractical as it directly refers to potentially (very) large set of maximal extensions of SP-structures. Below we present alternative definitions of SPS^{clo} and clo_{SPS}.

Definition 4 (SPC-structure)**.** *A* series-parallel closed structure *(or* SPC-*structure) is a structure* $spcs = \langle \Delta, \prec, \sqsubset \rangle$ *such that the following hold, for all* $x \neq y \in \Delta$*:*

- SPCS:1 $x \not\sqsubset x \not\prec x$
- SPCS:2 $x \prec y \implies y \not\sqsubset x$
- SPCS:3 $ISCS(spcs[x \longrightarrow y]) \neq \varnothing \implies y \sqsubset x$
- SPCS:4 $ISCS(spcs[x \dashrightarrow y]) \neq \varnothing \implies y \prec x$

The set of all SPC-*structures is denoted by* $SPCS$.

SPC-structures are positioned in-between SPM-structures and SP-structures.

Proposition 6. *$SPMS \subseteq SPCS \subseteq SPS$.*

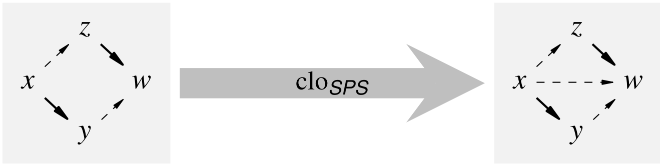

Figure 3: Closing SP-structure. Solid and dashed arcs represent relations $\prec \subseteq \sqsubset$ and $\sqsubset \setminus \prec$, respectively.

Proof. ($SPMS \subseteq SPCS$) Let $spms = \langle \Delta, \prec, \sqsubset \rangle \in SPMS$. Then SPCS:1 and SPCS:2 hold for $spms$, by SPMS:1 and SPMS:2, respectively. To show SPCS:3 for $spms$, suppose that $x \neq y \in \Delta$ and $ISCS(spms[x \longrightarrow y])$. Then, by Theorem 1, $x \not\prec y$. Hence, by SPMS:2,3, we obtain $y \sqsubset x$. Showing SPCS:4 for $spms$ is similar. As a result, $SPMS \subseteq SPCS$.

($SPCS \subseteq SPS$) By SPCS:1, it suffices to show that if $spcs = \langle \Delta, \prec, \sqsubset \rangle \in SPCS$ then we have $ISCS(spcs) = \varnothing$. Indeed, suppose that $\Phi \in ISCS(spcs)$. Then there are $x \neq y \in \Delta$ such that $x \prec y$. Since $spcs[x \longrightarrow y] = spcs$, by SPCS:3, we obtain $y \sqsubset x$, a contradiction with SPCS:2. ☐

To show that $SPCS = SPS^{clo}$, the next result identifies two ways in which SPCS-structures can be extended to SPS-structures.

Proposition 7. *Let $spcs = \langle \Delta, \prec, \sqsubset \rangle \in SPCS$ and $x \neq y \in \Delta$. Then:*

- $x \not\prec y \implies spcs[y \dashrightarrow x] \in SPS$.
- $x \not\sqsubset y \implies spcs[y \longrightarrow x] \in SPS$.

Proof. (1) If $spcs[y \dashrightarrow x] \notin SPS$, then $ISCS(spcs[y \dashrightarrow x]) \neq \varnothing$. Hence, by SPCS:4, we obtain $x \prec y$, yielding a contradiction.

(2) If $spcs[y \longrightarrow x] \notin SPS$, then $ISCS(spcs[y \longrightarrow x]) \neq \varnothing$. Hence, by SPCS:3, we obtain $x \sqsubset y$, yielding a contradiction. ☐

One can then define an alternative definition of the structure closure mapping for SP-structures. Let $\mathfrak{F} : SPS \to SPS$ be such that, for every $sps = \langle \Delta, \prec, \sqsubset \rangle \in SPS$, $\mathfrak{F}(sps) = \langle \Delta, \prec \cup \prec', \sqsubset \cup \sqsubset' \rangle$, where:

- $\prec' = \{\langle y, x \rangle \in \Delta \times \Delta \mid x \neq y \not\prec x \wedge ISCS(sps[x \dashrightarrow y]) \neq \varnothing\}$
- $\sqsubset' = \{\langle y, x \rangle \in \Delta \times \Delta \mid x \neq y \not\sqsubset x \wedge ISCS(sps[x \longrightarrow y]) \neq \varnothing\}$.

Such a mapping can be used to approximate the closure of an SP-structure sps and obtain it after repeatedly applying $\mathfrak{F}$ a bounded number of times. For example, Figure 3 depicts the SP-structure from Figure 2 together with its closure which can be calculated by repeated application of function $\mathfrak{F}$ as stated in the next result.

Proposition 8. *$\mathfrak{F}$ is a well-defined function such that, for every $sps \in SPS$:*

1. $sps \trianglelefteq \mathfrak{F}(sps) \trianglelefteq \mathrm{clo}_{SPS}(sps)$.
2. $\mathfrak{F}(sps) = sps$ *iff* $sps \in SPCS$.

Proof. (1) Suppose that $x \prec' y$. Then $x \neq y$ and $ISCS(sps[y \dashrightarrow x]) \neq \varnothing$. Hence, by Proposition 5(5), we obtain:

$$sps[x \longrightarrow y] \in SPS \text{ and } \max_{SPS}(sps[x \longrightarrow y]) = \max_{SPS}(sps).$$

Thus, since generating the same SPM-structures is equivalent to having the same closure, we have:

$$sps[x \longrightarrow y] \trianglelefteq \mathrm{clo}_{SPS}(sps[x \longrightarrow y]) = \mathrm{clo}_{SPS}(sps).$$

Similarly, we may show that $x \sqsubset' y$ implies

$$sps[x \dashrightarrow y] \trianglelefteq \mathrm{clo}_{SPS}(sps).$$

As result, we have $sps \trianglelefteq \mathfrak{F}(sps) \trianglelefteq \mathrm{clo}_{SPS}(sps)$. Thus $\mathfrak{F}(sps) \in SPS$, and so $\mathfrak{F}$ is well-defined.

(2) Follows directly from Definition 4 and the definition of $\mathfrak{F}$. $\square$

Proposition 8 reduces closure to finite saturation. Each application of $\mathfrak{F}$ preserves the intended semantics and adds only relations that must already hold in the closure, while fixed points are exactly structures belonging to *SPCS*. It remains only to observe that, over a finite domain, this process must stabilise after finitely many steps.

Theorem 2. *The mapping $\mathfrak{M} : SPS \to SPCS$ such that $\mathfrak{M}(sps) = \mathfrak{F}^{|\Delta_{sps}|^2}(sps)$, for every $sps \in SPS$, is the structure closure of SPS, i.e., $SPCS = SPS^{clo}$ and $\mathfrak{M} = \mathrm{clo}_{SPS}$.*

Proof. The number of relationships which the successive applications of $\mathfrak{F}$ can add is not greater than $|\Delta_{sps}|^2$ (taking into account all possible choices of two distinct event and at most two updates of binary relationships between them). Hence, $\mathfrak{F}(\mathfrak{M}(sps)) = \mathfrak{M}(sps)$, Hence, by Proposition 8(2), we obtain that $\mathfrak{M}(sps) \in SPCS$ and is $\mathfrak{M}$ is well-defined.

The theorem is proven by applying a general result — allowing one to deal with the closure operation and closed relational structures without referring to intersections of the maximal extensions of structures — formulated in [12] as Prop.7.8 and reproduced below.

> *Let $\mathscr{R}$ be an intersection-closed set of relational structures, $\mathscr{S} \subseteq \mathscr{R}$, and $f : \mathscr{R} \to \mathscr{S}$ be a monotonic (i.e., $rs \trianglelefteq rs' \implies f(rs) \trianglelefteq f(rs')$) and non-decreasing (i.e., $rs \trianglelefteq f(rs)$) function. Moreover, for all $rs = \langle \Delta, \prec, \sqsubset \rangle \in \mathscr{S}$ and $x \neq y \in \Delta$, we have:*
>
> - *$f(rs) \trianglelefteq rs$.*
> - *If $x \not\prec y$ (or $x \not\sqsubset y$), then there is $\overline{rs} \in \max_{\mathscr{R}}(rs)$ such that $x \not\prec_{\overline{rs}} y$ (resp. $x \not\sqsubset_{\overline{rs}} y$).*
>
> *Then f is the structure closure of $\mathscr{R}$, i.e., $\mathscr{S} = \mathscr{R}^{clo}$ and $f(rs) = \mathrm{clo}_{\mathscr{R}}(rs)$, for every $rs \in \mathscr{R}$.*

We can apply the above result after observing the following:

- *SPS* are intersection-closed.
- $SPCS \subseteq SPS$ by Proposition 6.
- $\mathfrak{M}$ is monotonic (i.e., $sps \trianglelefteq sps' \implies \mathfrak{M}(sps) \trianglelefteq \mathfrak{M}(sps')$) by Proposition 1.
- $\mathfrak{M}$ is non-decreasing (i.e., $sps \trianglelefteq \mathfrak{M}(sps)$) by Proposition 8(1).
- For every $spcs \in SPCS$, we have $\mathfrak{M}(spcs) \trianglelefteq spcs$, as $\mathfrak{M}(spcs) = spcs$ because in such a case $\prec' = \sqsubset' = \varnothing$ in the definition of $\mathfrak{F}$.
- For all $spcs = \langle \Delta, \prec, \sqsubset \rangle \in SPCS$ and $x \neq y \in \Delta$, if $x \not\prec y$ (or $x \not\sqsubset y$), then there is $spms \in \max_{SPS}(spcs)$ such that $x \not\prec_{spms} y$ (resp. $x \not\sqsubset_{spms} y$). This follows from Theorem 1 and Proposition 7.

Hence, $SPCS = SPS^{clo}$ and $\mathfrak{M} = \mathrm{clo}_{SPS}$. $\square$

As $\mathfrak{M}$ adds to *sps* all the arcs by all its maximal extensions, we obtained an effective procedure for determining the largest relational structure, with respect to extensions of relational structures, having a given set of maximal extensions.

7. Concluding remarks

This paper applied the generic approach elaborated in [11, 12] and developed an order-theoretic semantics for structured concurrency based on series-parallel orders. Specifications were represented as relational structures over sets of events equipped with two relations: *precedence*, expressing that one event occurs strictly 'earlier than' another, and *weak precedence*, expressing that one event is 'not later than' another.

We defined series-parallel structures as those relational structures that contain no indivisible strongly connected sets, showed that their maximal extensions coincide exactly with the relational encodings of series-parallel orders, and characterised closed series-parallel structures as intersections of their maximal extensions. We also introduced an iterative closure construction that provides an effective saturation procedure for non-maximal specifications while preserving the executions they generate. Taken together, these results yield a compositional framework for reasoning about causality and simultaneity in concurrent systems. The proposed framework is strictly more expressive than the Set-Sequential Object model and incomparable with the Interval-Sequential Object model.

A possible direction for the future work is concerned with the theory of traces [22]; in particular, the interval traces proposed and discussed in [23]. The relationship with interval relational structures from [13, 24] is yet to be developed.

Acknowledgments

This research was supported by the Leverhulme Trust grant RPG-2022-025, EPSRC grant EP/Y028813/1, and the Polish Minister of Science and Higher Education programme "Excellence Initiative - Research University". The authors are grateful to the anonymous referees for their comments.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] N. Wiener, A contribution to the theory of relative position, Proc. of the Cambridge Philosophical Society 33 (1914) 313–326.
[2] J. L. Gischer, Partial orders and the axiomatic theory of shuffle (pomsets), Ph.D. thesis, Stanford University, 1985.
[3] G. Boudol, I. Castellani, Concurrency and atomicity, Theor. Comput. Sci. 59 (1988) 25–84.
[4] L. Aceto, Full abstraction for series-parallel pomsets, in: Colloquium on Trees in Algebra and Programming, Springer, 1991, pp. 1–25.
[5] K. Lodaya, P. Weil, Series-parallel posets: algebra, automata and languages, in: Annual Symposium on Theoretical Aspects of Computer Science, Springer, 1998, pp. 555–565.
[6] K. Lodaya, P. Weil, Series–parallel languages and the bounded-width property, Theoretical Computer Science 237 (2000) 347–380.
[7] H. Plünnecke, K-density, N-density, and finiteness properties, in: Advances in Petri Nets 1984, Springer, 2005, pp. 392–412.
[8] D. Bechet, P. De Groote, C. Retoré, A complete axiomatisation for the inclusion of series-parallel partial orders, in: International Conference on Rewriting Techniques and Applications, Springer, 1997, pp. 230–240.
[9] R. Alur, C. Stanford, C. Watson, A robust theory of series parallel graphs, Proceedings of the ACM on Programming Languages 7 (2023) 1058–1088.
[10] U. Fahrenberg, C. Johansen, G. Struth, K. Ziemiański, Posets with interfaces as a model for concurrency, Information and Computation 285 (2022) 104914.
[11] R. Janicki, J. Kleijn, M. Koutny, Ł. Mikulski, Relational structures for concurrent behaviours, Theoretical Computer Science 862 (2021) 174–192.
[12] R. Janicki, J. Kleijn, M. Koutny, Ł. Mikulski, Paradigms of Concurrency - Observations, Behaviours, and Systems - a Petri Net View, volume 1020 of *Studies in Computational Intelligence*, Springer, 2022.
[13] R. Janicki, M. Koutny, Structure of concurrency, Theoretical Computer Science 112 (1993) 5–52.
[14] L. Lamport, The mutual exclusion problem: part I - a theory of interprocess communication, Journal of ACM 33 (1986) 313–326.
[15] E. Szpilrajn, Sur l'extension de l'ordre partiel, Fundam. Math. 16 (1930) 386–389.

[16] K. L. McMillan, Using unfoldings to avoid the state explosion problem in the verification of asynchronous circuits, in: CAV 1992, volume 663 of *Lecture Notes in Computer Science*, Springer, 1992, pp. 164–177.

[17] A. W. Mazurkiewicz, Trace theory, in: W. Brauer, W. Reisig, G. Rozenberg (Eds.), Petri Nets: Central Models and Their Properties, Advances in Petri Nets 1986, Part II, Proceedings of an Advanced Course, Bad Honnef, Germany, 8-19 September 1986, volume 255 of *Lecture Notes in Computer Science*, Springer, 1986, pp. 279–324.

[18] R. Janicki, J. Kleijn, M. Koutny, Ł. Mikulski, Relational structures for interval order semantics of concurrent systems, in: Petri Nets 2024, Lecture Notes in Computer Science, Springer, 2024, pp. 153–174.

[19] J. Kleijn, M. Koutny, Synchrony and asynchrony in membrane systems, in: 7th International Workshop of Membrane Computing, volume 4361 of *Lecture Notes in Computer Science*, Springer, 2006, pp. 66–85.

[20] A. Castañeda, S. Rajsbaum, M. Raynal, Specifying concurrent problems: beyond linearizability and up to tasks, in: DISC 2015, Springer, 2015, pp. 420–435.

[21] M. Koutny, L. Mikulski, Quasi-stratified order semantics of concurrency, CoRR abs/2407.13406 (2024). `arXiv:2407.13406`.

[22] V. Diekert, G. Rozenberg (Eds.), The Book of Traces, World Scientific, 1995.

[23] R. Janicki, X. Yin, Modeling concurrency with interval traces, Information and Computation 253 (2017) 78–108.

[24] R. Janicki, M. Koutny, Fundamentals of modelling concurrency using discrete relational structures, Acta Informatica 34 (1997) 367–388.