Pomsets with Boxes: Protection, Separation, and Locality in Concurrent Kleene Algebra

Paul Brunet, David Pym

Pomsets with Boxes: Protection, Separation, and Locality in Concurrent Kleene Algebra

Paul Brunet and David Pym

University College London
{p.brunet,d.pym}@ucl.ac.uk

Abstract. Concurrent Kleene Algebra is an elegant tool for equational reasoning about concurrent programs. An important feature of concurrent programs that is missing from CKA is the ability to restrict legal interleavings. To remedy this we extend the standard model of CKA, namely pomsets, with a new feature, called boxes, which can specify that part of the system is protect from outside interference. We study the algebraic properties of this new model. Another drawback of CKA is that the language used for expressing properties of programs is the same as that which is used to express programs themselves. This is often too restrictive for practical purposes. We provide a logic, 'pomset logic', that is an assertion language for specifying such properties, and which is interpreted on pomsets with boxes. We develop the basic metatheory for the relationship between pomset logic and CKA and illustrate this relationship with simple examples.

Keywords: Concurrent Kleene Algebra· Pomsets· Atomicity· Semantics· Separation· Local reasoning· Bunched logic.

1 Introduction

Concurrent Kleene Algebra (CKA) [8] is an elegant tool for equational reasoning about concurrent programs. However, the language used for expressing properties of programs is the same as that which is used to express programs themselves.

It is clear that this situation is not ideal for specifying and reasoning about properties of programs. Any language specifiable in CKA terms has bounded width (i.e., the number of processes in parallel; the size of a maximal independent set) and bounded depth (i.e., the number of alternations of parallel and sequential compositions). However, any property involving the existence of a pattern — e.g., the set of pomsets satisfying a given property — has both unbounded width and unbounded depth.

In this paper, we provide a logic, 'pomset logic', that is an assertion language for specifying such properties. We develop the basic metatheory for the relationship between pomset logic and CKA and illustrate this relationship with simple examples. In addition, to the usual classical or intuitionistic connectives — both

are possible — the logic includes connectives that characterise both sequential and parallel composition.

In addition, we note that CKA allows programs with every possible interleaving of parallel threads. However, to prove the correctness of such programs, some restrictions must be imposed on what are the legal interleavings. We provide a mechanism of 'boxes' for this purpose. Boxes identify protected parts of the system, so restricting the possible interleavings. From the point of view of the outside of the box, the behaviour of the box is as though the program within box be atomic. However, boxes can be nested, with this atomicity observation holding at each level. Pomset logic has context and box modalities that characterise this situation.

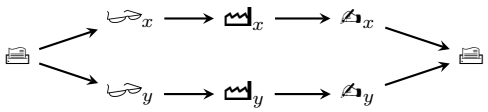
Example 1 (Running example: a distributed counter).

We consider here a program where a counter is incremented in parallel by two processes. The intention is that the counter should be incremented twice, once by each process. However, to do so each process has to first load the contents of the counter, then compute the increment, and finally commit the result to memory. A naive implementation is presented in Table 1.

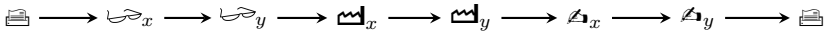
	`print(counter);`	
`x:=counter;`	‖	`y:=counter;`
`x:=x+1;`	‖	`y:=y+1;`
`counter:=x;`	‖	`counter:=y;`
	`print(counter);`	

Table 1: Distributed counter

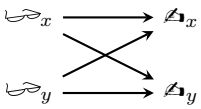
To get simpler pictures, we represent the print instruction `print(counter)` by 🖶, the read instruction `x:=counter` by read_x, the increment instruction `x:=x+1` by inc_x, and finally the write instruction `counter:=x` by write_x. Doing so, we may represent the program from Table 1 as follows:



This program does not comply with our intended semantics: indeed a possible run of this program goes as follows:



The result is that the counter has been incremented by one. We can identify a subset of instructions that indicate there is a fault: the problem is that both read instructions happened before both write instructions, i.e.:

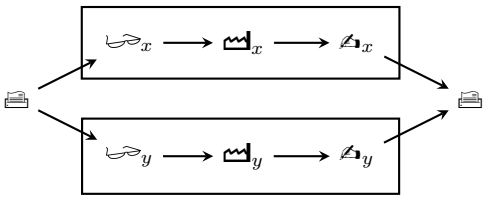


To fix this problem, a simple solution is to make the sequence "read;compute;write" *atomic*. This yields the program in Table 2.

```
                print(counter);
atomic{         ||  atomic{
  x:=counter;   ||    y:=counter;
  x:=x+1;       ||    y:=y+1;
  counter:=x;   ||    counter:=y;
}               ||  }
                print(counter);
```

Table 2: Distributed counter with atomic increment

Diagrammatically, this can be represented by drawing solid boxes around the `atomic{}` blocs:



With these boxes, it is no longer possible for problematic behaviour we described earlier to happen. We will show in this paper how to make this formal.

The paper has a number of remaining sections. In Section 2, we extend pomsets with a new construct for protection, namely boxes. We provide a syntax for specifying such pomsets and characterise precisely its expressivity. This enables us, for example, to correctly represent the program from Table 1. We axiomatize the equational theory of this model.

In Section 3, we introduce pomset logic. This logic comes in both classical and intuitionistic variants. In addition to the usual classical or intuitionistic connectives, this logic includes connectives corresponding to each of sequential and parallel composition. These two classes of connectives are combined to give the overall logics in the same way as BI's additives and multiplicatives [14,1,16]. The logics also include modalities that characterise respectively protection and locality. These correspondences are made precise by a van Benthem–Hennessy–Milner-type theorem asserting that two programs are (operationally) equivalent iff they satisfy the same formulae.

Finally, in Section 4, we briefly discuss how local reasoning principles can be established for our framework of programs and their logics.

For space considerations, some proofs are deferred to the appendix.

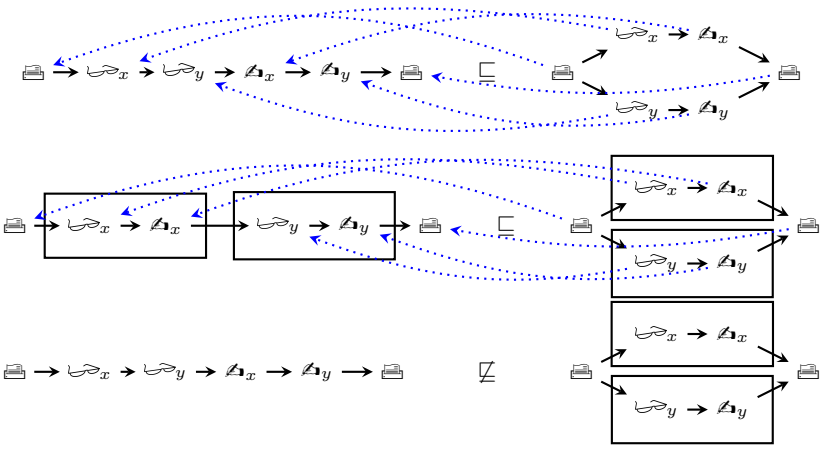


Fig. 1: Poset subsumption

2 Algebra of Pomsets with Boxes

In this section we define our semantic model, and the corresponding syntax. We characterise the expressivity of the syntax, and axiomatise its equational theory. The results in this section have been fully formalised in Coq; the development is available on GitHub: `https://github.com/monstrencage/AtomicCKA`.

For the remainder of this section, we fix an set Σ of atomic actions.

2.1 Pomsets with boxes

Definition 1 (Poset with boxes). *A* poset with boxes *is given by a tuple* $P := \langle \mathcal{E}_P, \leq_P, \lambda_P, \mathcal{B}_P \rangle$*, where* $\mathcal{E}_P$ *is a finite set of* events*;* $\leq_P \subseteq \mathcal{E}_P \times \mathcal{E}_P$ *is a partial order;* $\lambda_P : \mathcal{E}_P \to \Sigma$ *is a labelling function;* $\mathcal{B}_P \subseteq \mathcal{P}(\mathcal{E}_P)$ *is a set of* boxes*, such that* $\emptyset \notin \mathcal{B}_P$*.*

Definition 2 (Poset morphisms). *A* (poset with boxes) homomorphism *is a map between event-sets that is bijective, label respecting, order preserving, and box preserving. In other words, a map* $\phi : \mathcal{E}_P \to \mathcal{E}_Q$ *such that (i)* ϕ *is a bijection; (ii)* $\lambda_Q \circ \phi = \lambda_P$*; (iii)* $\phi(\leq_P) \subseteq \leq_Q$*; (iv)* $\phi(\mathcal{B}_P) \subseteq \mathcal{B}_Q$*. If in addition (iii) holds as an equality,* ϕ *is called* order-reflecting*. If on the other hand (iv) holds as an equality* ϕ *is* box-reflecting*. A homomorphism that is both order- and box-reflecting is a* (poset with boxes) isomorphism*.*

In Figure 1 are some examples and a non-example of subsumption between posets. We introduce some notations. $\mathbb{P}_\Sigma$ is the set of posets with boxes. If ϕ is a homomorphism from P to Q, we write $\phi : P \to Q$. If there exists such a homomorphism (respectively an isomorphism) from P to Q, we write $Q \sqsubseteq P$ (resp. $Q \cong P$).

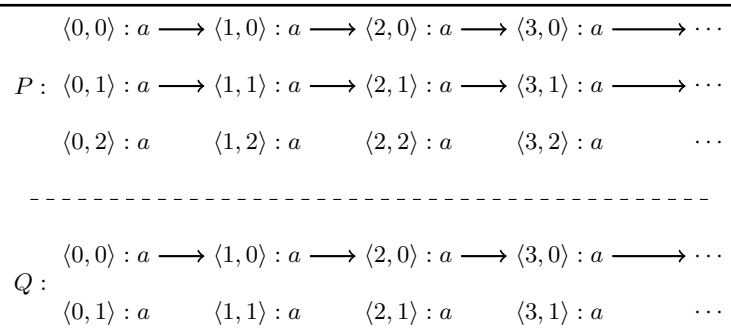


Fig. 2: Example of mutual homomorphic pomsets that are not isomorphic

Lemma 1. *$\cong$ is an equivalence relation. $\sqsubseteq$ is a partial order with respect to $\cong$.*

Remark 1. Note that the fact that $\sqsubseteq$ is antisymmetric with respect to $\cong$ relies on the finiteness of the posets considered here. Indeed, we can build infinite pomsets that are not isomorphic but have nevertheless homomorphism between them in both directions. An example is provided in Figure 2.

Definition 3 (Pomsets with boxes). Pomsets with boxes *are equivalence classes of $\cong$. The set $\mathbf{Pom}_\Sigma$ of pomsets with boxes is defined as $\mathbb{P}_\Sigma/_\cong$.*

We now define some elementary operations to build posets.

Definition 4 (Constants). *Given a symbol $a \in \Sigma$, the* atomic poset *associated with a is defined as $\mathbb{a} := \langle \{0\}, [0 \mapsto a], Id_{\{0\}}, \emptyset \rangle \in \mathbb{P}_\Sigma$.*

The empty poset is defined as $\mathbb{e} := \langle \emptyset, \emptyset, \emptyset, \emptyset \rangle \in \mathbb{P}_\Sigma$.

Remark 2. For any poset $P \in \mathbb{P}_\Sigma$, $P \sqsubseteq \mathbb{e} \Leftrightarrow P \sqsupseteq \mathbb{e} \Leftrightarrow P \cong \mathbb{e}$. This is because each of those relations imply there is a bijection between the events of P and $\mathcal{E}_\mathbb{e} = \emptyset$. So we know that P has no events, and since boxes cannot be empty, P has no boxes either. Hence $P \cong \mathbb{e}$.

Definition 5 (Compositions). *Let P, Q be two posets with boxes. The sequential composition $P \otimes Q$ and parallel composition $P \parallel Q$ are defined by:*

$$P \oplus Q := \langle \mathcal{E}_P \uplus \mathcal{E}_Q, \leq_P \cup \leq_Q, \lambda_P \sqcup \lambda_Q, \mathcal{B}_P \cup \mathcal{B}_Q \rangle$$
$$P \otimes Q := \langle \mathcal{E}_P \uplus \mathcal{E}_Q, \leq_P \cup \leq_Q \cup (\mathcal{E}_P \times \mathcal{E}_Q), \lambda_P \sqcup \lambda_Q, \mathcal{B}_P \cup \mathcal{B}_Q \rangle$$

where the symbol $\sqcup$ denotes the union of two functions, i.e. given $f : A \to C$ and $g : B \to C$, the function $f \sqcup g : A \uplus B \to C$ associates $f(a)$ to $a \in A$ and $g(b)$ to $b \in B$.

Intuitively, $P \oplus Q$ consists of disjoint copies of P and Q side by side. $P \otimes Q$ also contains disjoint copies of P and Q, but also considers every event in P to be smaller that any event in Q.

Definition 6 (Boxing). *Given a pomset P its* boxing *is denoted by $[P]$ and is defined by:* $[P] := \langle \mathcal{E}_P, \leq_P, \lambda_P, \mathcal{B}_P \cup \{\mathcal{E}_P\} \rangle$.

Boxing a pomset simply amounts to drawing a box around it.

Definition 7 (Restriction, sub-poset). *For a set of events $A \subseteq \mathcal{E}_P$, we may define the* restriction of P to A *as:*

$$P\restriction_A := \langle A, \leq_P \cap (A \times A), \lambda_P\restriction_A, \mathcal{B}_P \cap \mathcal{P}(A) \rangle .$$

We say that P is a sub-poset *of Q, and write $P \Subset Q$, if there is a set $A \subseteq \mathcal{E}_Q$ such that $P \cong Q\restriction_A$.*

Definition 8 (Pomset terms, SP-Pomsets). *A* (pomset) term *is a syntactic expression generated from the following grammar:*

$$s, t \in \mathtt{SP}_\Sigma ::= 1 \mid a \mid s\,;t \mid s \parallel t \mid [s] .$$

By convention ; *binds tighter than* $\parallel$. *A term is interpreted as a poset as follows:*

$$\begin{array}{lll} [\![a]\!] := \mathfrak{a} & [\![1]\!] := \mathfrak{e} & [\![[s]]\!] := [[\![s]\!]] \\ [\![s\,;t]\!] := [\![s]\!] \otimes [\![t]\!] & [\![s \parallel t]\!] := [\![s]\!] \oplus [\![t]\!] & \end{array}$$

A pomset $[P]_\cong$ is called series-parallel *if it is the interpretation of some term, i.e. $\exists s \in \mathtt{SP}_\Sigma : [\![s]\!] \cong P$.*

Example 2. The program in Table 1 of the running example corresponds to:

$$[\![\text{🖶} ; (\text{🖱}_x ; \text{🖬}_x ; \text{✍}_x \parallel \text{🖱}_y ; \text{🖬}_y ; \text{✍}_y) ; \text{🖶}]\!] .$$

The corrected program, from Table 2, corresponds to:

$$[\![\text{🖶} ; ([\text{🖱}_x ; \text{🖬}_x ; \text{✍}_x] \parallel [\text{🖱}_y ; \text{🖬}_y ; \text{✍}_y]) ; \text{🖶}]\!] .$$

Finally, the problematic pattern we identified may be represented as:

$$[\![(\text{🖱}_x \parallel \text{🖱}_y) ; (\text{✍}_x \parallel \text{✍}_y)]\!] .$$

Sets of posets We now lift our operations and relations to sets of posets.

Definition 9 (Orderings on sets of posets). *Let $A, B \subseteq \mathbb{P}_\Sigma$ be two sets of posets. We define the following relations:*

Isomorphic inclusion $A \lesssim B$ *iff* $\forall P \in A, \exists Q \in B : P \cong Q$.
Isomorphic equivalence $A \cong B$ *iff* $A \lesssim B \wedge B \lesssim A$.
Subsumption $A \sqsubseteq B$ *iff* $\forall P \in A, \exists Q \in B : P \sqsubseteq Q$.

Remark 3. Isomorphic inclusion and subsumption are a partial orders with respect to isomorphic equivalence, which is an equivalence relation.

Definition 10 (Operations on sets of posets). *We will use the set-theoretic union of sets of posets, as well as the pointwise liftings of the two products of posets and the boxing operators:*

$$A \otimes B := \{P \otimes Q \mid \langle P, Q\rangle \in A \times B\} \qquad [A] := \{[P] \mid P \in A\}$$
$$A \oplus B := \{P \oplus Q \mid \langle P, Q\rangle \in A \times B\}.$$

Definition 11 (Closure of a set of posets). *The* (downwards) closure *of a set of posets S is the smallest set containing S that is downwards closed with respect to the subsumption order, i.e.: $S{\downarrow} := \{P \in \mathbb{P}_\Sigma \mid \exists Q \in S : P \sqsubseteq Q\}$. Similarly, the* upwards closure *of S is defined as: $S{\uparrow} := \{P \in \mathbb{P}_\Sigma \mid \exists Q \in S : P \sqsupseteq Q\}$.*

Remark 4. $(\_){\downarrow}$ and $(\_){\uparrow}$ are Kuratowski closure operators [11], i.e. they satisfy the following properties:

$$\emptyset{\downarrow} = \emptyset \qquad A \subseteq A{\downarrow} \qquad A{\downarrow}{\downarrow} = A{\downarrow} \qquad (A \cup B){\downarrow} = A{\downarrow} \cup B{\downarrow}.$$

(And similarly for the upwards closure.) Using downwards-closures, we may express subsumption in terms of isomorphic inclusion:

$$A \sqsubseteq B \qquad \Leftrightarrow \qquad A \subsetneq B{\downarrow} \qquad \Leftrightarrow \qquad A{\downarrow} \subsetneq B{\downarrow}.$$

Similarly, the equivalence relation associated with $\sqsubseteq$, defined as the intersection of the relation and its converse, corresponds to the predicate $A{\downarrow} \cong B{\downarrow}$.

Definition 12. *Terms are defined by the following grammar:*

$$e, f \in \mathtt{T}_\Sigma ::= 0 \mid 1 \mid a \mid e\,;f \mid e \parallel f \mid e + f \mid [e]$$

The can be interpreted as finite sets of posets with boxes as follows:

$$[\![0]\!] := \emptyset \qquad [\![1]\!] := \{\mathfrak{e}\} \qquad [\![a]\!] := \{\mathfrak{a}\}$$

$$[\![[e]]\!] := [[\![e]\!]] \qquad [\![e\,;f]\!] := [\![e]\!] \otimes [\![f]\!] \qquad [\![e + f]\!] := [\![e]\!] \cup [\![f]\!] \qquad [\![e \parallel f]\!] := [\![e]\!] \oplus [\![f]\!].$$

Remark 5. Interpreted as programs, 0 represents failure: this is a program that aborts the whole execution. + on the other hand represents non-deterministic choice. It can be used to model conditional branching.

2.2 A characterisation of series-parallel pomsets

Theorem 1. *A pomset $[P]_\cong$ is series-parallel iff and only if it does not contain any of the patterns in Figure 3, i.e. none of the following properties are satisfied:*

- $\boldsymbol{P}_1$: $\exists e_1, e_2, e_3, e_4 \in \mathcal{E}_P : e_1 \leq_P e_3 \wedge e_2 \leq_P e_3 \wedge e_2 \leq_P e_4 \wedge e_1 \not\leq_P e_4 \wedge e_2 \not\leq_P e_1 \wedge e_4 \not\leq_P e_3$.
- $\boldsymbol{P}_2$: $\exists e_1, e_2, e_3 \in \mathcal{E}_P, \exists A, B \in \mathcal{B}_P : e_1 \in A \setminus B \wedge e_2 \in A \cap B \wedge e_3 \in B \setminus A$.
- $\boldsymbol{P}_3$: $\exists e_1, e_2, e_3 \in \mathcal{E}_P, \exists A \in \mathcal{B}_P : e_1 \notin A \wedge e_2, e_3 \in A \wedge e_1 \leq_P e_2 \wedge e_1 \not\leq_P e_3$.
- $\boldsymbol{P}_4$: $\exists e_1, e_2, e_3 \in \mathcal{E}_P, \exists A \in \mathcal{B}_P : e_1 \notin A \wedge e_2, e_3 \in A \wedge e_2 \leq_P e_1 \wedge e_3 \not\leq_P e_1$.

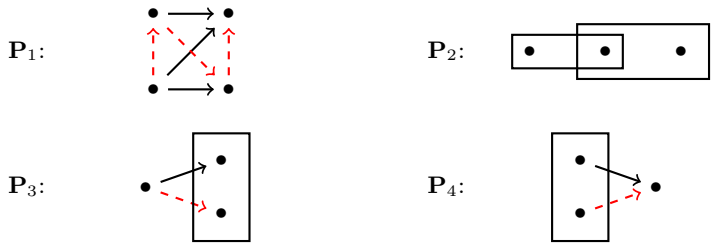


Fig. 3: Forbidden patterns of SP-pomsets: dashed arrows (in red) are negated.

Before we discuss the proof of this result, we make a number of comments.

The four properties in Theorem 1 are invariant under isomorphism, since they only use the ordering between events and the membership of events to boxes. This is consistent with SP being a property of pomsets, not posets.

Pattern $\mathbf{P}_1$ is known as N, and is the forbidden pattern of series-parallel pomsets (without boxes), as proved by Gischer [5]. Pattern $\mathbf{P}_2$ indicates that the boxes in an SP-pomset are well nested: two boxes are either disjoint, or one is contained in the other. Patterns $\mathbf{P}_3$ and $\mathbf{P}_4$ reflect that if an event e is outside of a box B, then e cannot distinguish events in B by the order: e can be smaller than all of B, larger than all of B, or incomparable with every event in B.

Together, $\mathbf{P}_2$, $\mathbf{P}_3$, and $\mathbf{P}_4$ provide an alternative view of pomsets with boxes: one may see them as *hyper-pomsets*, i.e. pomsets where some events (the boxes) can be labelled with non-empty pomsets (the contents of the boxes). However, even though this definition is equivalent to the one we use, it seems that for our purposes the definition we provide is more convenient. In particular, the definition of hyper-pomset homomorphism is more involved.

Proof (Sketch). By a simple induction on terms, we can easily show that sp-posets avoid all four forbidden patterns. The more challenging direction is the converse: given a poset P that does not contain any of the forbidden patterns, can we build a term $s \in \mathtt{SP}_\Sigma$ such that $P \cong [\![s]\!]$. We construct this by induction on the size of P, defined as number of boxes plus the number of events. Notice that if a poset does not contain a pattern, any sub-poset does not either.

If P has at most one event, then the following property holds:

- if P has no events, then $P \cong [\![1]\!]$;
- if P has a single event e, let $a = P](e)$; we know that

$$\mathcal{B}_P \subseteq \{\beta \subseteq \{e\} \mid \beta \neq \emptyset\} = \{\{e\}\}$$

  - if $\mathcal{B}_P = \emptyset$ then $P \cong \mathbb{o} = [\![a]\!]$;
  - if $\mathcal{B}_P = \{\{e\}\}$ then $P \cong [\mathbb{o}] = [\![[a]]\!]$.

If $\mathcal{E}_P \in \mathcal{B}_P$, then let P' be the poset obtained by removing the box $\mathcal{E}_P$. The size of P' is strictly smaller than that of P, and $P \cong [P']$. By induction we get a term s such that $[\![s]\!] \cong P'$, so $P \cong [\![[s]]\!]$.

$$\frac{e = f \in A}{A \vdash e = f} \qquad A \vdash e = e \qquad \frac{A \vdash e = f}{A \vdash f = e} \qquad \frac{A \vdash e = f \qquad A \vdash f = g}{A \vdash e = g}$$

$$\sigma, \tau : \Sigma \to \mathtt{T}_\Sigma, \ \frac{\forall a \in \Sigma,\ A \vdash \sigma(a) = \tau(a)}{A \vdash \hat{\sigma}(e) = \hat{\tau}(e).}$$

$$\frac{e = f \in A}{A \vdash e \leq f} \qquad \frac{f = e \in A}{A \vdash e \leq f} \qquad \frac{e \leq f \in A}{A \vdash e \leq f} \qquad A \vdash e \leq e \qquad \frac{A \vdash e \leq f \qquad A \vdash f \leq g}{A \vdash e \leq g}$$

$$\sigma, \tau : \Sigma \to \mathtt{T}_\Sigma, \ \frac{\forall a \in \Sigma,\ A \vdash \sigma(a) \leq \tau(a)}{A \vdash \hat{\sigma}(e) \leq \hat{\tau}(e).}$$

Table 3: Equational and inequational logic

Consider now a pomset P with at least two events, and such that $\mathcal{E}_P \notin \mathcal{B}_P$. A set of events $A \subseteq \mathcal{E}_P$ is called:

- **non-trivial** if $A \notin \{\emptyset, \mathcal{E}_P\}$;
- **nested** if for any box $\beta \in \mathcal{B}_P$ either $\beta \subseteq A$ or $A \cap \beta = \emptyset$;
- **prefix** if for any $e \in A$ and $f \notin A$ we have $e \leq_P f$;
- **isolated** if for any $e \in A$ and $f \notin A$ we have $e \not\leq_P f$ and $f \not\leq_P e$.

If P contains a non-trivial, nested, prefix set A, then we may conclude by induction, since: $P \cong P\restriction_A \otimes P\restriction_{\overline{A}}$. If P contains a non-trivial, nested, isolated set A, then we may conclude by induction, since: $P \cong P\restriction_A \oplus P\restriction_{\overline{A}}$. To conclude, it suffices to show the following pair of claims (whose proofs are omitted):

(Claim 1) if P contains a non-trivial prefix set, it contains one that is nested;
(Claim 2) if P contains a non-trivial isolated set, it contains one that is nested.

Indeed, as a corollary of Gischer's characterisation theorem [5, Theorem 3.1], we know that since P is N-free (i.e. does not contain $\mathbf{P}_1$) and contains at least two events, it contains either a non-trivial prefix set or a non-trivial isolated set. □

2.3 Axiomatic presentations of pomset algebra

We now introduce axioms to capture the various order and equivalence relations we introduced over posets and sets of posets. Given a set of axioms A (i.e. universally quantified identities), we write $A \vdash e = f$ to denote that the pair $\langle e, f \rangle$ belongs to the smallest congruence containing every axiom in A. Equivalently, $A \vdash e = f$ holds iff this statement is derivable in equational logic, as described in Table 3. Similarly, $A \vdash e \leq f$ is the smallest precongruence containing A, where equality axioms are understood as pairs of inequational axioms. An inference

$s\,;(t\,;u) = (s\,;t)\,;u$	(A1)	$e+(f+g) = (e+f)+g$	(B1)
$s \parallel (t \parallel u) = (s \parallel t) \parallel u$	(A2)	$e+f = f+e$	(B2)
$s \parallel t = t \parallel s$	(A3)	$e+e = e$	(B3)
$1\,;s = s$	(A4)	$0+e = e$	(B4)
$s\,;1 = s$	(A5)	$0\,;e = e\,;0 = 0$	(B5)
$1 \parallel s = s$	(A6)	$0 \parallel e = 0$	(B6)
$[[s]] = [s]$	(A7)	$e\,;(f+g) = (e\,;f)+(e\,;g)$	(B7)
$[1] = 1.$	(A8)	$(e+f)\,;g = (e\,;g)+(f\,;g)$	(B8)
		$e \parallel (f+g) = (e \parallel f)+(e \parallel g)$	(B9)
$(s \parallel t)\,;(u \parallel v) \leq (s\,;u) \parallel (t\,;v)$	(C1)	$[0] = 0$	(D1)
$[s] \leq s.$	(C2)	$[e+f] = [e]+[f]$	(D2)
		$(s \parallel t)\,;(u \parallel v)+(s\,;u) \parallel (t\,;v) = (s\,;u) \parallel (t\,;v)$	(E1)
		$[s]+s = s.$	(E2)

Table 4: Axioms

system is also provided in Table 3. We will consider the following sets of axioms:

$$\begin{aligned}
\mathrm{BiMon}_\Box &:= (\mathrm{A1}) - (\mathrm{A8}) && \text{(Bimonoid with boxes)}\\
\mathrm{CMon}_\Box &:= \mathrm{BiMon}_\Box, (\mathrm{C1}), (\mathrm{C2}) && \text{(Concurrent monoid with boxes)}\\
\mathrm{SR}_\Box &:= \mathrm{BiMon}_\Box, (\mathrm{B1}) - (\mathrm{B9}) && \text{(Bisemiring with boxes)}\\
\mathrm{CSR}_\Box &:= \mathrm{CSR}_\Box, (\mathrm{E1}), (\mathrm{E2}). && \text{(Concurrent semiring with boxes)}
\end{aligned}$$

Remark 6. One can show that for $A \in \{\mathrm{SR}_\Box, \mathrm{CSR}_\Box\}$, we have:

$$A \vdash e \leq f \Leftrightarrow A \vdash e+f = f \qquad A \vdash e = f \Leftrightarrow A \vdash e \leq f \wedge A \vdash f \leq e.$$

Posets up to isomorphisms: the free bimonoid In this section, we prove the following soundness and completeness theorem:

Theorem 2. $[\![s]\!] \cong [\![t]\!] \Leftrightarrow \mathrm{BiMon}_\Box \vdash s = t.$

The key ingredient in this proof is the construction of a syntactic version of the restriction operator, which extracts a subpomsets out of a pomsets, guided by a subset of events.

Definition 13 (Syntactic restriction). *Let $s \in \mathtt{SP}_\Sigma$ be a term and $A \subseteq \mathcal{E}_{[\![s]\!]}$ a set of events. The* syntactic restriction *of s to A, written $\pi_A(s)$ is defined by*

induction on terms as follows:

$$\pi_A(1) := 1 \qquad \pi_A(s;t) := \pi_{A\cap\mathcal{E}_{[\![s]\!]}}(s);\pi_{A\cap\mathcal{E}_{[\![s]\!]}}(t)$$

$$\pi_A(a) := \begin{cases} 1 \; if \; A = \emptyset \\ a \; otherwise \end{cases} \qquad \pi_A(s \parallel t) := \pi_{A\cap\mathcal{E}_{[\![s]\!]}}(s) \parallel \pi_{A\cap\mathcal{E}_{[\![s]\!]}}(t).$$

$$\pi_A([s]) := \begin{cases} [s] \qquad if \; A = \mathcal{E}_{[\![e]\!]} \\ \pi_A(s) \; otherwise \end{cases}$$

The main properties of this operator are stated in the following lemma:

Lemma 2. *For any $s \in \mathtt{SP}_\Sigma$ and $A \subseteq \mathcal{E}_{[\![s]\!]}$ the following holds:*

(i) $[\![\pi_A(s)]\!] \cong [\![s]\!]\downarrow_A$;
(ii) if A is prefix and nested (see Section 2.2 for definitions), then

$$\mathrm{BiMon}_\Box \vdash \pi_A(s);\pi_{\overline{A}}(s) = s;$$

(iii) if A is isolated and nested (see Section 2.2 for definitions), then

$$\mathrm{BiMon}_\Box \vdash \pi_A(s) \parallel \pi_{\overline{A}}(s) = s.$$

(Here $\overline{A}$ denotes the complement of A relative to $\mathcal{E}_{[\![s]\!]}$, i.e. $\overline{A} := \mathcal{E}_{[\![e]\!]} \setminus A$.)

The proof of Theorem 2 then follows by induction on terms.

Posets up to subsumption: the free concurrent monoid

Theorem 3. $[\![s]\!] \sqsubseteq [\![t]\!] \Leftrightarrow \mathrm{CMon}_\Box \vdash s \leq t$

To prove this result, we will deal with the two extra axioms (C1) and (C2) separately. In order to do so, we define the following sub-orders of $\sqsubseteq$:

Box-subsumption We write $P \sqsubseteq_b Q$ if there is an order-reflecting poset homomorphism $\phi : Q \to P$.
Order-subsumption We write $P \sqsubseteq_o Q$ if there is an box-reflecting poset homomorphism $\phi : Q \to P$.

Lemma 3 (Factorisation of subsumption). *If $P \sqsubseteq Q$, then there are R_1, R_2 such that: $P \sqsubseteq_o R_1 \sqsubseteq_b Q$ and $P \sqsubseteq_b R_2 \sqsubseteq_o Q$. In other words, $\sqsubseteq = \sqsubseteq_o \circ \sqsubseteq_b$ and $\sqsubseteq = \sqsubseteq_b \circ \sqsubseteq_o$.*

Proof. Let $\phi : Q \to P$ be a poset homomorphism witnessing $P \sqsubseteq Q$. We may define $R_1 := \langle \mathcal{E}_Q, \leq_Q, \lambda_Q, \{B \mid \phi(B) \in \mathcal{B}_P\}\rangle$ and $R_2 := \langle \mathcal{E}_P, \leq_P, \lambda_P, \phi(\mathcal{B}_Q)\rangle$. Checking that $P \sqsubseteq_o R_1 \sqsubseteq_b Q$ and $P \sqsubseteq_b R_2 \sqsubseteq_o Q$ hold is a simple matter of unfolding definitions. □

We also notice the following properties of $\sqsubseteq_b$ and $\sqsubseteq_o$ with respect to the forbidden patterns of series parallel posets:

Lemma 4. *Let P, Q be two posets:*

(i) if $P \sqsubseteq_b Q$*, then* P *contains* $\boldsymbol{P}_1$ *iff* Q *contains* $\boldsymbol{P}_1$*;*
(ii) if $P \sqsubseteq_o Q$*, then* P *contains* $\boldsymbol{P}_2$ *iff* Q *contains* $\boldsymbol{P}_2$*;*
(iii) if $P \sqsubseteq_b Q$ *and* Q *contains* $\boldsymbol{P}_3$*, then* P *contains* $\boldsymbol{P}_3$*;*
(iv) if $P \sqsubseteq_b Q$ *and* Q *contains* $\boldsymbol{P}_4$*, then* P *contains* $\boldsymbol{P}_4$*.*

We now prove the first half of the theorem:

Lemma 5. *If* $[\![s]\!] \sqsubseteq_b [\![t]\!]$*, then* $\mathrm{CMon}_\square \vdash s \leq t$*.*

Proof. First, we define the following operator $\mathcal{H}(\_) : \mathbb{P}_\Sigma \to \mathcal{P}_f(\mathbb{P}_\Sigma)$:

$$\mathcal{H}(P) := \{\langle \mathcal{E}_P, \leq_P, \lambda_P, B\rangle \mid B \subseteq \mathcal{B}_P\}.$$

From the definitions it is straightforward to show that $P \sqsubseteq_b Q$ iff Q is isomorphic to some $P' \in \mathcal{H}(P)$. This operator $\mathcal{H}(P)$ can be mirrored on terms, i.e. we can associate inductively to each term s a finite set of terms $\mathrm{H}(s)$ such that $\forall P \in \mathcal{H}([\![s]\!]), \exists t \in \mathrm{H}(s) : [\![t]\!] \cong P$ and $\forall t \in \mathrm{H}(s), \mathrm{CMon}_\square \vdash s \leq t$. We may therefore conclude:

$$\begin{aligned}
[\![s]\!] \sqsubseteq_b [\![t]\!] &\Rightarrow \exists P \in \mathcal{H}([\![s]\!]) : [\![t]\!] \cong P \\
&\Rightarrow \exists t' \in \mathrm{H}(s) : [\![t']\!] \cong [\![t]\!] \\
&\Rightarrow \exists t' \in \mathrm{H}(s) : \mathrm{BiMon}_\square \vdash t' = t \wedge \mathrm{CMon}_\square \vdash s \leq t' \\
&\Rightarrow \mathrm{CMon}_\square \vdash s \leq t' = t.
\end{aligned}$$

□

We now prove the second half of the theorem:

Lemma 6. *If* $[\![s]\!] \sqsubseteq_o [\![t]\!]$ *then* $\mathrm{CMon}_\square \vdash s \leq t$*.*

Remark 7. The proof we give below relies on Gischer's completeness theorem [5]. In the Coq proof however, we make no assumptions, and we do not have access to Gischer's result. Therefore we perform a different, more technically involved proof there, with Gischer's theorem as a corollary.

Proof. We will perform this proof by induction on the number of boxes in s. Let $\phi : [\![t]\!] \to [\![s]\!]$ be the box-reflecting homomorphism witnessing $[\![s]\!] \sqsubseteq_o [\![t]\!]$.

If s contains no boxes, then by Gischer's completeness theorem we know that

$$[\![s]\!] \sqsubseteq_o [\![t]\!] \Rightarrow \mathrm{BiMon}_\square, (\mathrm{C}_1) \vdash s \leq t.$$

Hence, as we have $\mathrm{BiMon}_\square, (\mathrm{C}_1) \subseteq \mathrm{CMon}_\square$, we get $\mathrm{CMon}_\square \vdash s \leq t$.

If on the other hand s has boxes, consider the following set of boxes:

$$\mathcal{B} := \left\{B \in \mathcal{B}_{[\![s]\!]} \;\middle|\; \forall C \in \mathcal{B}_{[\![s]\!]}, B \subseteq C \Rightarrow B = C\right\} \left(= \max \mathcal{B}_{[\![s]\!]}\right).$$

Notice that since ϕ is box-reflecting, we have that $\mathcal{B} = \phi\left(\max \mathcal{B}_{[\![t]\!]}\right)$. Furthermore, we have the following property: for any box $B \in \max \mathcal{B}_{[\![t]\!]}$, the map $\phi\downarrow_B$ is a box-reflecting homomorphism from $[\![t]\!]\downarrow_B$ to $[\![s]\!]\downarrow_{\phi(B)}$. We pick new symbols for the elements of $\mathcal{B}$, i.e. we find a set Σ' disjoint from Σ and a bijection

$\ell(\_) : \mathcal{B} \to \Sigma'$. Using the observation (A.4) we made earlier, we find two maps $s(\_), t(\_) : \Sigma'$ such that:

$$\forall B \in \mathcal{B}, \qquad \mathrm{BiMon}_\Box \vdash \pi_B(s) = [s(\ell(B))]. \tag{2.1}$$

$$\forall x \in \Sigma', \qquad \mathcal{E}_{[\![s(x)]\!]} \notin \mathcal{B}_{[\![s(x)]\!]}. \tag{2.2}$$

$$\forall B \in \max \mathcal{B}_{[\![t]\!]}, \qquad \mathrm{BiMon}_\Box \vdash \pi_B(t) = [t(\ell(\phi(B)))]. \tag{2.3}$$

$$\forall x \in \Sigma', \qquad \mathcal{E}_{[\![t(x)]\!]} \notin \mathcal{B}_{[\![t(x)]\!]}. \tag{2.4}$$

Clearly, for every $x \in \Sigma'$, $s(x)$ has strictly less boxes than s. Our previous observations imply that $[\![s(x)]\!] \sqsubseteq_o [\![t(x)]\!]$. Therefore, by induction hypothesis, we get that $\forall x \in \Sigma'$, $\mathrm{CMon}_\Box \vdash s(x) \leq t(x)$, hence:

$$\forall x \in \Sigma', \mathrm{CMon}_\Box \vdash [s(x)] \leq [t(x)].$$

Combined with (2.1) and (2.3) this yields:

$$\forall B \in \max \mathcal{B}_{[\![t]\!]}, \mathrm{CMon}_\Box \vdash \pi_{\phi(B)}(s) \leq \pi_B(t).$$

We define two substitutions $\sigma, \tau : \Sigma \cup \Sigma' \to \mathtt{SP}_\Sigma$ as follows:

$$\sigma(x) := \begin{cases} \pi_B(s) & \text{if } x = \ell(B) \\ x & \text{if } x \in \Sigma \end{cases} \qquad \tau(x) := \begin{cases} \pi_B(t) & \text{if } x = \ell(\phi(B)) \\ x & \text{if } x \in \Sigma \end{cases}$$

Finally, we syntactically substitute maximal boxed subterms with letters from Σ' in s, t, yielding terms $s', t' \in \mathtt{SP}_{\Sigma \cup \Sigma'}$ such that $\hat{\sigma}(s') = s$, $\hat{\tau}(t') = t$, and neither s' nor t' has any box. By unfolding the definitions, we can check that since $[\![s]\!] \sqsubseteq_o [\![t]\!]$ we have $[\![s]\!]' \sqsubseteq_o [\![t]\!]'$. Furthermore, since s' and t' do not contain any box, we may use Gischer's theorem to prove that $\mathrm{CMon}_\Box \vdash s' \leq t'$. We may now conclude: by applying σ everywhere in the proof of $\mathrm{CMon}_\Box \vdash s' \leq t'$, we get that $\mathrm{CMon}_\Box \vdash s = \hat{\sigma}(s') \leq \hat{\sigma}(t')$. Since $\forall a \in \Sigma \cup \Sigma'$ we have $\mathrm{CMon}_\Box \vdash \sigma(a) \leq \tau(a)$, we get $\mathrm{CMon}_\Box \vdash \hat{\sigma}(t') \leq \hat{\tau}(t') = t$. □

Finite sets of posets as free algebras The following lemma allows us to extend seamlessly our completeness theorem from $\mathrm{BiMon}_\Box$ to $\mathrm{SR}_\Box$ and from $\mathrm{CMon}_\Box$ to $\mathrm{CSR}_\Box$.

Lemma 7. *There is a function $T_\_ : \mathtt{T}_\Sigma \to \mathcal{P}_f(\mathtt{SP}_\Sigma)$ such that:*

$$\mathrm{SR}_\Box \vdash e = \sum_{s \in T_e} s. \qquad [\![e]\!] \cong \{[\![s]\!] \mid s \in T_e\}.$$

Proof. Simple induction on terms. □

Corollary 1. $[\![e]\!] \cong [\![f]\!] \Leftrightarrow \mathrm{SR}_\Box \vdash e = f$

Corollary 2. $[\![e]\!]{\downarrow} \cong [\![f]\!]{\downarrow} \Leftrightarrow \mathrm{CSR}_\Box \vdash e = f$.

3 Logic for pomsets with boxes

We introduce a logic for reasoning about pomsets with boxes. The logic we introduce is a bunched modal logic, in the sense of [14,6,4,1,16], with substructural connectives corresponding to each of sequential and concurrent composition. Modalities characterise boxes and locality. The logic is also conceptually related to Concurrent Separation Logic [13,3].

3.1 Pomset logic: definitions

We generate the set of formulas $\mathtt{F}_\Sigma$ and the set of positive formulas $\mathtt{F}^+_\Sigma$ as follows:

$$\phi, \psi \in \mathtt{F}^+_\Sigma ::= \bot \mid a \mid \phi \vee \psi \mid \phi \wedge \psi \mid \phi \blacktriangleright \psi \mid \phi \star \psi \mid [\phi] \mid (\!|\phi|\!)$$
$$\phi, \psi \in \mathtt{F}_\Sigma ::= \bot \mid a \mid \phi \vee \psi \mid \phi \wedge \psi \mid \phi \blacktriangleright \psi \mid \phi \star \psi \mid [\phi] \mid (\!|\phi|\!) \mid \neg\phi$$

Remark 8. Here the atomic predicates are chosen to be exactly Σ. Another natural choice would be a separate set $Prop$ of atomic predicates, together with a valuation $v : Prop \to \mathcal{P}(\Sigma)$ to indicate which actions satisfy which predicate. It is equivalent:

- to encode a formula over $Prop$ as a formula over Σ, simply replace every predicate $p \in Prop$ with the formula $\bigvee_{a \in v(p)} a$
- to encode a formula over Σ as one over $Prop$, we need to make the customary assumption that $\forall a \in \Sigma$, $\exists p \in Prop : v(p) = \{a\}$.

These formulas are interpreted over posets. We define two satisfaction relations: $\models_K$ allows one to reason on posets modulo *isomorphism*, while $\models_{J\uparrow}$ and $\models_{J\downarrow}$ consider them up to *subsumption*:

$P \models_K \phi$ is defined by induction on $\phi \in \mathtt{F}_\Sigma$:
- $P \models_K \bot$ iff $P \cong \mathbb{e}$
- $P \models_K a$ iff $P \cong \mathbb{a}$
- $P \models_K \neg\phi$ iff $P \not\models_K \phi$
- $P \models_K \phi \vee \psi$ iff $P \models_K \phi$ or $P \models_K \psi$
- $P \models_K \phi \wedge \psi$ iff $P \models_K \phi$ and $P \models_K \psi$
- $P \models_K \phi \blacktriangleright \psi$ iff $P \cong P_1 \otimes P_2$ such that $P_1 \models_K \phi$ and $P_2 \models_K \psi$
- $P \models_K \phi \star \psi$ iff $P \cong P_1 \oplus P_2$ such that $P_1 \models_K \phi$ and $P_2 \models_K \psi$
- $P \models_K [\phi]$ iff $P \cong [Q]$ and $Q \models_K \phi$
- $P \models_K (\!|\phi|\!)$ iff $P \Subset Q$ and $Q \models_K \phi$.

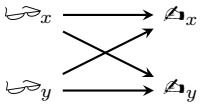
$P \models_{J\uparrow} \phi$ is defined by induction on $\phi \in \mathtt{F}^+_\Sigma$:
- $P \models_{J\uparrow} \bot$ iff $P \cong \mathbb{e}$
- $P \models_{J\uparrow} a$ iff $P \sqsupseteq \mathbb{a}$
- $P \models_{J\uparrow} \phi \vee \psi$ iff $P \models_{J\uparrow} \phi$ or $P \models_{J\uparrow} \psi$
- $P \models_{J\uparrow} \phi \wedge \psi$ iff $P \models_{J\uparrow} \phi$ and $P \models_{J\uparrow} \psi$
- $P \models_{J\uparrow} \phi \blacktriangleright \psi$ iff $P \sqsupseteq P_1 \otimes P_2$ such that $P_1 \models_{J\uparrow} \phi$ and $P_2 \models_{J\uparrow} \psi$
- $P \models_{J\uparrow} \phi \star \psi$ iff $P \sqsupseteq P_1 \oplus P_2$ such that $P_1 \models_{J\uparrow} \phi$ and $P_2 \models_{J\uparrow} \psi$

- $P \models_{J\uparrow} [\phi]$ iff $P \sqsupseteq [Q]$ and $Q \models_{J\uparrow} \phi$
- $P \models_{J\uparrow} (\!|\phi|\!)$ iff $P \sqsupseteq P' \Rsh Q$ and $Q \models_{J\uparrow} \phi$.

$P \models_{J\downarrow} \phi$ is defined by induction on $\phi \in \mathtt{F}^+_\Sigma$:

- $P \models_{J\downarrow} \bot$ iff $P \cong \mathbb{c}$
- $P \models_{J\downarrow} a$ iff $P \sqsubseteq \mathbb{a}$
- $P \models_{J\downarrow} \phi \vee \psi$ iff $P \models_{J\downarrow} \phi$ or $P \models_{J\downarrow} \psi$
- $P \models_{J\downarrow} \phi \wedge \psi$ iff $P \models_{J\downarrow} \phi$ and $P \models_{J\downarrow} \psi$
- $P \models_{J\downarrow} \phi \blacktriangleright \psi$ iff $P \sqsubseteq P_1 \otimes P_2$ such that $P_1 \models_{J\downarrow} \phi$ and $P_2 \models_{J\downarrow} \psi$
- $P \models_{J\downarrow} \phi \star \psi$ iff $P \sqsubseteq P_1 \oplus P_2$ such that $P_1 \models_{J\downarrow} \phi$ and $P_2 \models_{J\downarrow} \psi$
- $P \models_{J\downarrow} [\phi]$ iff $P \sqsubseteq [Q]$ and $Q \models_{J\downarrow} \phi$
- $P \models_{J\downarrow} (\!|\phi|\!)$ iff $P \sqsubseteq P' \Rsh Q$ and $Q \models_{J\downarrow} \phi$.

Example 3. Recall the problematic pattern we saw in the running example, i.e.



This pattern can be represented by the following formula:

$$conflict := (\!|(\curvearrowright_x \star \curvearrowright_y) \blacktriangleright (\mathbb{a}_x \star \mathbb{a}_y)|\!).$$

We may also interpret these formulas over sets of posets. We consider here two ways a set of posets X may satisfy a formula:

- X satisfies ϕ *universally* if every poset in X satisfies ϕ;
- X satisfies ϕ *existentially* if some poset in X satisfies ϕ.

Combined with our three satisfaction relations for individual pomsets, this yields six definitions:

$$X \models^{\forall}_{K} \phi \text{ iff } \forall P \in X, P \models_K \phi \qquad X \models^{\exists}_{K} \phi \text{ iff } \exists P \in X, P \models_K \phi$$
$$X \models^{\forall}_{J\uparrow} \phi \text{ iff } \forall P \in X, P \models_{J\uparrow} \phi \qquad X \models^{\exists}_{J\uparrow} \phi \text{ iff } \exists P \in X, P \models_{J\uparrow} \phi$$
$$X \models^{\forall}_{J\downarrow} \phi \text{ iff } \forall P \in X, P \models_{J\downarrow} \phi \qquad X \models^{\exists}_{J\downarrow} \phi \text{ iff } \exists P \in X, P \models_{J\downarrow} \phi.$$

For a term $e \in \mathtt{T}_\Sigma$, we write $e \models^y_x \phi$ to mean $[\![e]\!] \models^y_x \phi$.

3.2 Some facts about pomset logic

Lemma 8. *For every formula $\phi \in \mathtt{F}^+_\Sigma$, if $P \models_K \phi$ then $P \models_{J\uparrow} \phi$ and $P \models_{J\downarrow} \phi$.*

Proof. This is straightforward, since $\models_K$ is obtained from $\models_{J\uparrow}$ by replacing $\sqsupseteq$ with $\cong$, and since $P \cong Q \Rightarrow P \sqsupseteq Q$. A similar argument holds for $\models_{J\downarrow}$. □

Lemma 9. *For a formula $\Phi \in \mathtt{F}_\Sigma$ and a pair of posets $P, Q \in \mathbb{P}_\Sigma$, if $P \cong Q$ then $P \models_K \Phi$ iff $Q \models_K \Phi$.*

Proof. We proceed by induction on Φ.

- If $\Phi = \bot$, $P \models_K \bot$ means that $\mathfrak{e} \cong P \cong Q$, i.e. $Q \models_K \bot$.
- If $\Phi = \neg\phi$, $\Phi = \phi \vee \psi$, or $\Phi = \phi \wedge \psi$, we use the induction hypothesis to show that $P \models_K \Phi$ iff $Q \models_K \Phi$.
- If $\Phi = a$, $\Phi = \phi \blacktriangleright \psi$, $\Phi = \phi \star \psi$, $\Phi = [\phi]$, or $\Phi = (\!|\phi|\!)$, then the satisfaction relation says $P \models_K \Phi$ iff $P \cong P'$ and $h(P')$. Since $Q \cong P$, Q also satisfies $Q \cong P'$ and $h(P')$ so we get $Q \models_K \Phi$ without using the induction hypothesis. By symmetry, the converse implication holds as well. □

In other words, two isomorphic posets K-satisfy the same formulas. This means K-satisfiability is defined on pomsets.

Lemma 10. *For a formula $\Phi \in \mathtt{F}^+_\Sigma$ and a pair of posets $P, Q \in \mathbb{P}_\Sigma$, if $P \sqsubseteq Q$ then the following hold: $P \models_{J\uparrow} \Phi \Rightarrow Q \models_{J\uparrow} \Phi$, and $Q \models_{J\downarrow} \Phi \Rightarrow P \models_{J\downarrow} \Phi$.*

We can build formulas from series-parallel terms: $\phi(a) := a$, $\phi(1) := \bot$, $\phi([s]) := [\phi(s)]$, $\phi(s\,;t) := \phi(s) \blacktriangleright \phi(t)$, and $\phi(s \parallel t) := \phi(s) \star \phi(t)$.

Lemma 11. *For any sp-term s and any poset P, we have:*

$$P \models_K \phi(s) \Leftrightarrow P \cong [\![s]\!] \qquad P \models_{J\uparrow} \phi(s) \Leftrightarrow P \sqsupseteq [\![s]\!] \qquad P \models_{J\downarrow} \phi(s) \Leftrightarrow P \sqsubseteq [\![s]\!]\,.$$

3.3 Algebraic presentation

Given a formula ϕ and a satisfaction relation $\models_X$, with $X \in \{J\uparrow, J\downarrow, K\}$, we may define the X-semantics of ϕ as $[\![\phi]\!]_X := \{P \in \mathbb{P}_\Sigma \mid P \models_X \phi\}$. We may rephrase the results of the previous subsection from this perspective:

Lemma 8 : $[\![\phi]\!]_K \subseteq [\![\phi]\!]_{J\uparrow} \cap [\![\phi]\!]_{J\downarrow}$.
Lemma 9 : $[\![\phi]\!]_K$ is closed under $\cong$.
Lemma 10 : $[\![\phi]\!]_{J\uparrow} = [\![\phi]\!]_{J\uparrow}\uparrow$ and $[\![\phi]\!]_{J\downarrow} = [\![\phi]\!]_{J\downarrow}\downarrow$, i.e. the $J\uparrow$ semantics is upwards-closed and the $J\downarrow$-semantics is downwards-closed.
Lemma 11 : $[\![\phi(s)]\!]_K \cong \{[\![s]\!]\}$, $[\![\phi(s)]\!]_{J\downarrow} \cong \{[\![s]\!]\}\downarrow$, $[\![\phi(s)]\!]_{J\uparrow} \cong \{[\![s]\!]\}\uparrow$.

We may also use this to describe the satisfaction relations for sets of posets:

$$e \models^{\exists}_X \phi \Leftrightarrow [\![e]\!] \cap [\![\phi]\!]_X \neq \emptyset \qquad\qquad e \models^{\forall}_X \phi \Leftrightarrow [\![e]\!] \subseteq [\![\phi]\!]_X\,. \tag{3.1}$$

With these definitions, we make the following observations:

Lemma 12. *The following statements hold universally:*

$$e \models^{\forall}_K \phi \Leftrightarrow [\![e]\!] \subsetneq [\![\phi]\!]_K \tag{3.2}$$
$$e \models^{\forall}_{J\downarrow} \phi \Leftrightarrow [\![e]\!] \sqsubseteq [\![\phi]\!]_{J\downarrow} \tag{3.3}$$
$$e \models^{\forall}_{J\uparrow} \phi \Leftrightarrow \forall P \in [\![e]\!]\,,\ \exists Q \in [\![\phi]\!]_{J\uparrow} :\ P \sqsupseteq Q. \tag{3.4}$$
$$e \subsetneq f \Rightarrow \forall \phi,\, e \models^{\exists}_K \phi \Rightarrow f \models^{\exists}_K \phi \tag{3.5}$$
$$e \sqsubseteq f \Rightarrow \forall \phi,\, e \models^{\exists}_{J\uparrow} \phi \Rightarrow f \models^{\exists}_{J\uparrow} \phi \tag{3.6}$$
$$e \subsetneq f \Rightarrow \forall \phi,\, f \models^{\forall}_K \phi \Rightarrow e \models^{\forall}_K \phi \tag{3.7}$$
$$e \sqsubseteq f \Rightarrow \forall \phi,\, f \models^{\forall}_{J\downarrow} \phi \Rightarrow e \models^{\forall}_{J\downarrow} \phi \tag{3.8}$$
$$e \models^{\exists}_K \phi(s) \Leftrightarrow [\![s]\!] \in [\![e]\!] \tag{3.9}$$
$$e \models^{\exists}_{J\uparrow} \phi(s) \Leftrightarrow [\![s]\!] \in [\![e]\!]\downarrow \tag{3.10}$$

We may use $\phi(\_)$ and $T_{\_}$ to generate formulas out of terms: let $e \in \mathtt{T}_\Sigma$ be a term, we define the formula $\Phi(e) := \bigvee_{s \in T_e} \phi(s)$.

Lemma 13. *$\Phi(e)$ satisfies the following equivalences:*

$$e \models^{\forall}_K \Phi(f) \Leftrightarrow [\![e]\!] \subsetneq [\![f]\!] \tag{3.11}$$

$$e \models^{\forall}_{J\downarrow} \Phi(f) \Leftrightarrow [\![e]\!] \sqsubseteq [\![f]\!] \tag{3.12}$$

Proof. Recall that since $\_\uparrow$ and $\_\downarrow$ are Kuratowski closure operators, they distribute over unions. We may thus obtain:

$$[\![\Phi(e)]\!]_K = \bigcup_{s \in T_e} [\![\phi(s)]\!]_K \cong \bigcup_{s \in T_e} \{[\![s]\!]\} \cong [\![e]\!]\,.$$

$$[\![\Phi(e)]\!]_{J\downarrow} = \bigcup_{s \in T_e} [\![\phi(s)]\!]_{J\downarrow} \cong \bigcup_{s \in T_e} \{[\![s]\!]\}\downarrow \cong [\![e]\!]\downarrow.$$

The statements then follow by (3.2) and (3.3). □

3.4 Adequacy of pomset logic

In this section, we present adequacy lemmas. These should be understood as appropriate formulations of the completeness theorems relating operational equivalence and logical equivalence in the sense of van Benthem [2] and Hennessy–Milner [7,12] for this logic (cf. [1]). From the results we have established so far, we may directly prove the following:

Proposition 1. *For a pair of series parallel terms s, t, the identity* $\mathrm{BiMon}_\Box \vdash s = t$ *is derivable if and only if for any formula ϕ, $[\![s]\!] \models_K \phi$ iff $[\![t]\!] \models_K \phi$.*

Proposition 2. *For a pair of terms $s, t \in \mathtt{SP}_\Sigma$, the identity* $\mathrm{CMon}_\Box \vdash s \leq t$ *is derivable if and only if for any formula ϕ, $[\![s]\!] \models_{J\uparrow} \phi$ implies $[\![t]\!] \models_{J\uparrow} \phi$.*

Proposition 3. *Given two terms $e, f \in \mathtt{T}_\Sigma$, the following equivalences hold:*

$$\mathrm{SR}_\Box \vdash e \leq f \Leftrightarrow \left(\forall \phi,\, e \models^{\exists}_K \phi \Rightarrow f \models^{\exists}_K \phi\right) \Leftrightarrow \left(\forall \phi,\, f \models^{\forall}_K \phi \Rightarrow e \models^{\forall}_K \phi\right) \tag{3.13}$$

$$\mathrm{SR}_\Box \vdash e = f \Leftrightarrow \left(\forall \phi,\, e \models^{\exists}_K \phi \Leftrightarrow f \models^{\exists}_K \phi\right) \Leftrightarrow \left(\forall \phi,\, e \models^{\forall}_K \phi \Leftrightarrow f \models^{\forall}_K \phi\right). \tag{3.14}$$

Proof. (3.13) We prove both directions:

(⇒) Assume $\mathrm{SR}_\Box \vdash e \leq f$. By Corollary 1 this means $[\![e]\!] \subsetneq [\![f]\!]$. Therefore, we may conclude by (3.5) and (3.7).

(⇐) We show that each LHS implies $[\![e]\!] \subsetneq [\![f]\!]$, i.e. $\mathrm{SR}_\Box \vdash e \leq f$:

- Assume $\forall \phi,\, f \models^{\forall}_K \phi \Rightarrow e \models^{\forall}_K \phi$. Then in particular, since $[\![f]\!] \subsetneq [\![f]\!]$ by Lemma 13 we have $f \models^{\forall}_K \Phi(f)$, hence $e \models^{\forall}_K \Phi(f)$ ergo $[\![e]\!] \subsetneq [\![f]\!]$.
- Assume $\forall \phi,\, e \models^{\exists}_K \phi \Rightarrow f \models^{\exists}_K \phi$, and let $P \in [\![e]\!]$. By Lemmas 7 we know that there is $s \in T_e$ such that $P \cong [\![s]\!]$, and by Lemma 11 we get $e \models^{\exists}_K \phi(s)$, hence $f \models^{\exists}_K \phi(s)$. Hence by (3.9) we get $P \cong [\![s]\!] \in [\![f]\!]$.

(3.14) follows from (3.13), and the fact that $\leq$ is antisymmetric. □

We may also prove the variants of this proposition for $J{\downarrow}$ and $J{\uparrow}$:

Proposition 4. *Given two terms $e, f \in \mathtt{T}_\Sigma$, the following equivalences hold:*

$$\mathrm{CSR}_\Box \vdash e \leq f \Leftrightarrow \left(\forall \phi,\, e \models^{\exists}_{J\uparrow} \phi \Rightarrow f \models^{\exists}_{J\uparrow} \phi\right) \tag{3.15}$$

$$\mathrm{CSR}_\Box \vdash e = f \Leftrightarrow \left(\forall \phi,\, e \models^{\exists}_{J\uparrow} \phi \Leftrightarrow f \models^{\exists}_{J\uparrow} \phi\right). \tag{3.16}$$

Proposition 5. *Given two terms $e, f \in \mathtt{T}_\Sigma$, the following equivalences hold:*

$$\mathrm{CSR}_\Box \vdash e \leq f \Leftrightarrow \left(\forall \phi,\, f \models^{\forall}_{J\downarrow} \phi \Rightarrow e \models^{\forall}_{J\downarrow} \phi\right) \tag{3.17}$$

$$\mathrm{CSR}_\Box \vdash e = f \Leftrightarrow \left(\forall \phi,\, e \models^{\forall}_{J\downarrow} \phi \Leftrightarrow f \models^{\forall}_{J\downarrow} \phi\right). \tag{3.18}$$

4 Local Reasoning

The discussion in this section does not rely on which satisfaction relation we pick, so in the remainder we assume we picked some relation $\models \in \{\models_K, \models_{J\uparrow}, \models_{J\uparrow}\}$.

The modality $(\!|\bot|\!)$ provides an explicit way of performing local reasoning (in the sense of (Concurrent) Separation Logic [10,17,13,3]). Indeed, if a program satisfies some formula ϕ, then, if we insert this program in any context, the resulting program will satisfy $(\!|\phi|\!)$. (Relationships between Concurrent Separation Logic and CKA have bee explored in [15].)

Pomset logic enjoys a high level of compositionality, much like algebraic logic. Formally, this comes from the following principle:

$$\text{If } e \models \phi \text{ and } \forall a,\, \sigma a \models \tau a, \text{ then } \hat{\sigma} e \models \hat{\tau}\phi.$$

This makes possible the following verification scenario. Let P be a large program, involving a number of simpler sub-programs $p_1, \ldots, p_n$. We may simplify P by replacing the sub-programs by uninterpreted symbols $x_1, \ldots, x_n$. We then check that this simplified program satisfies a formula Φ, the statement of which might involve the x_i. We then separately determine for each sub-program p_i some specification ϕ_i. Finally, using the principle we just stated, we can show that the full program P satisfies the formula Φ', obtained by replacing the x_i with ϕ_i.

To illustrate how the box modality relates to protection, consider the classical formula $(\!|[\neg(\!|a|\!)]|\!)$. A poset satisfies this formula iff it contains a box that do not contain any a-labelled event. Note that the negation is used here to express a closed-world property, which is not intrinsically classical, but cannot be expressed in our intuitionistic language. If a program p satisfies $(\!|[\neg(\!|a|\!)]|\!)$, then any program featuring p positively also satisfies it. This can be understood as saying that if p contains a correct box, then no use of p can introduce an a-bug inside that box, justifying our claim that boxes embody protection.

Acknowledgements

This work has been supported by UK EPRSC Research Grant EP/R006865/1: Interface Reasoning for Interacting Systems (IRIS). The authors are grateful to their colleagues at UCL and within the IRIS project for their interest.

References

1. G. Anderson and D. Pym. A calculus and logic of bunched resources and processes. *Theoretical Computer Science* 614:63-96, 2016.
2. J. van Benthem. *Logical Dynamics of Information and Interaction.* Cambridge University Press, 2014.
3. S. Brookes and P. O'Hearn. Concurrent Separation Logic. *ACM SIGLOG News* 3(3), 47–65, 2016.
4. M. Collinson and D. Pym. Algebra and Logic for Resource-based Systems Modelling. *Mathematical Structures in Computer Science* 19:959–1027, 2009. doi:10.1017/S0960129509990077.
5. Gischer, J.L.: The equational theory of pomsets. Theor. Comput. Sci. **61**(2-3), 199–224 (1988). https://doi.org/10.1016/0304-3975(88)90124-7
6. D. Galmiche, D. Méry, and D. Pym. The semantics of BI and resource tableaux. *Math. Str. Comp. Sci.*, 15(06):1033–1088, 2005.
7. M. Hennessy and G. Plotkin. On observing nondeterminsm and concurrency. *Proc. 7th ICALP*. LNCS 85:299–309, 1980.
8. Hoare, T., Möller, B., Struth, G., Wehrman, I.: Concurrent Kleene algebra. In: CONCUR. pp. 399–414 (2009). https://doi.org/10.1007/978-3-642-04081-8_27
9. Tony Hoare, Bernhard Möller, Georg Struth, Ian Wehrman. Concurrent Kleene Algebra and its Foundations. *Journal of Logic and Algebraic Programming* 80, 266–296, 2011.
10. S. Ishtiaq and P. O'Hearn. BI as an assertion language for mutable data structures. *Proceedings of the 28th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, 14–26, 2001.
11. Kuratowski, C.: Sur l'opération Ā de l'Analysis Situs. Fundamenta Mathematicae **3**(1), 182–199 (1922)
12. R. Milner. *Communication and Concurrency.* Prentice Hall, 1989.
13. P. O'Hearn. Resources, concurrency, and local reasoning. *Theoretical Computer Science* 375 (1–3), 2007, 271–307.
14. P. O'Hearn and D. Pym. The logic of bunched implications. *Bull. Symb. Log.*, 5(2):215–244, 1999.
15. P. O'Hearn, R. L. Petersen, J. Villard and A. Hussain. On the relation between Concurrent Separation Logic and Concurrent Kleene Algebra. *Journal of Logical and Algebraic Methods in Programming* 84(3), 285–302, 2015.
16. D. Pym. Resource Semantics: Logic as a Modelling Technology. ACM SIGLOG News, April 2019, Vol. 6, No. 2, 5–41.
17. J. Reynolds. Separation Logic: a logic for shared mutable data structures. In *Proc LICS '02*, IEEE Comp. Soc. Press, 55–74 2002.

A Omitted proofs

Proof (Claim 1). assume there exists non-trivial prefix set. We pick a minimal one, i.e. a non-trivial prefix set A such that for any non-trivial prefix B, if $B \subseteq A$, then $B = A$ (this is always possible since $\subseteq$ is a well-founded partial order on finite sets). If A is nested, then A satisfies our requirements. Otherwise, there is a box that is not contained in A while intersecting A. Since P does not contain the pattern $\mathbf{P}_2$, we know that we may pick a maximal such box β. This means that we know the following:

$$\beta \in \mathcal{B}_P \qquad (\forall \alpha \in \mathcal{B}_P,\, \beta \subseteq \alpha \Rightarrow \beta = \alpha) \qquad \exists e_1 \in \beta \cap A \qquad \exists e_2 \in \beta \setminus A.$$

First, we show that $A \subseteq \beta$. Consider the set $A' := A \setminus \beta$. Clearly $A' \subsetneq A$ (since $e_1 \in A \setminus A'$). We may also show that A' is prefix. Let $e \in A'$ and $f \notin A'$. There are two cases:

- either $f \notin A$, then since A is prefix and $e \in A' \subseteq A$ we have $e \leq_P f$;
- or $f \in A \cap \beta$. In this case, we use the fact that P does not have pattern $\mathbf{P}_3$: we know that $e_2 \in \beta \setminus A$, so since $e \in A' \subseteq A$ we have $e \leq_P e_2$, and since $e \notin \beta$ and $e_2, f \in \beta$ we may conclude that $e \leq_P f$.

Therefore A' is prefix and strictly contained in A. By minimality of A, A' has to be trivial. Since A is non-trivial this means that $A' = \emptyset$, hence that $A \subseteq \beta$.

Now we know that $A \subseteq \beta$. Because we know that β is not empty, and that $\mathcal{E}_P \notin \mathcal{B}_P$, we deduce that β is non-trivial. Since P does not contain pattern $\mathbf{P}_2$, and by maximality of β, we know that β is nested. We now conclude by showing that β is in fact prefix. Let $e \in \beta$, and $f \notin \beta$. Since $A \subseteq \beta$ we get that $f \notin A$. By the prefix property of A we get $e_1 \leq_P f$, and since $e_1, e \in \beta$ and $f \notin \beta$, by the absence of pattern $\mathbf{P}_4$ we get that $e \leq_P f$. □

Proof (Claim 2). This proceeds in a similar manner as Claim 1. We pick a minimal non-trivial isolated set A, and try to find a maximal box β such that $\beta \cap A \neq \emptyset$ and $\beta \setminus A \neq \emptyset$. If no such box exists, A is already nested. If we do find such a box, we first show that A has to be contained in β. Then we use this to show that β is a non-trivial nested isolated set.

As noted before, this concludes the proof, Claim 1 and Claim 2 being enough to call on our inductive hypothesis and build a term out of the poset. □

Theorem 2. $[\![s]\!] \cong [\![t]\!] \Leftrightarrow \mathrm{BiMon}_\square \vdash s = t$.

Before proving this theorem, we need to check the following statements:

$$\mathcal{E}_{[\![t]\!]} = \emptyset \Rightarrow \mathrm{BiMon}_\square \vdash t = 1 \tag{A.1}$$

$$\mathcal{E}_{[\![s]\!]} \in \mathcal{B}_{[\![s]\!]} \Rightarrow \exists t :\ \mathrm{BiMon}_\square \vdash s = [t] \wedge \mathcal{E}_{[\![t]\!]} \notin \mathcal{B}_{[\![t]\!]} \tag{A.2}$$

Proof (statement (A.3)*).* Since $\mathcal{E}_{[\![t]\!]} = \emptyset$, t does not feature any symbol from Σ, meaning $t \in \mathtt{SP}_\emptyset$. We may then show that for every term $t \in \mathtt{SP}_\emptyset$ we have $\mathrm{BiMon}_\square \vdash t = 1$. □

Proof (statement (A.4)*).* We proceed by induction on s:

$s = 1$, $s = a$ contradicts the premise;
$s = [s']$ we have to consider two cases:
- $\mathcal{E}_{[\![s']\!]} \in \mathcal{B}_{[\![s']\!]}$: in this case, by induction we get t such that $\mathcal{E}_{[\![t]\!]} \notin \mathcal{B}_{[\![t]\!]}$, and $\mathrm{BiMon}_\Box \vdash s = [s'] = [[t]] = [t]$.
- $\mathcal{E}_{[\![s']\!]} \notin \mathcal{B}_{[\![s']\!]}$: pick $t = s'$.

$s = s_1 \,;\, s_2$ we know the following facts:

$$\mathcal{E}_{[\![s_1]\!]} \cup \mathcal{E}_{[\![s_2]\!]} = \mathcal{E}_{[\![s]\!]} \in \mathcal{B}_{[\![s]\!]} = \mathcal{B}_{[\![s_1]\!]} \cup \mathcal{B}_{[\![s_2]\!]}$$
$$\forall \beta \in \mathcal{B}_{[\![s_i]\!]},\ \beta \subseteq \mathcal{E}_{[\![s_i]\!]}$$
$$\mathcal{E}_{[\![s_1]\!]} \cap \mathcal{E}_{[\![s_2]\!]} = \emptyset$$

From them we deduce that either $\mathcal{E}_{[\![s_1]\!]} = \emptyset$ or $\mathcal{E}_{[\![s_2]\!]} = \emptyset$. We conclude by applying the induction hypothesis to the appropriate subterm, and use (A.3) to conclude.

$s = s_1 \parallel s_2$ same as $s_1 \,;\, s_2$. □

We will also need the following observations:

$$\mathcal{E}_{[\![t]\!]} = \emptyset \Rightarrow \mathrm{BiMon}_\Box \vdash t = 1 \tag{A.3}$$
$$\mathcal{E}_{[\![s]\!]} \in \mathcal{B}_{[\![s]\!]} \Rightarrow \exists t :\ \mathrm{BiMon}_\Box \vdash s = [t] \wedge \mathcal{E}_{[\![t]\!]} \notin \mathcal{B}_{[\![t]\!]} \tag{A.4}$$

We may now prove Theorem 2:

Proof (Theorem 2). As often for this kind of result, the right-to-left implication, i.e. soundness, is very simple to check, by a simple induction on the derivation.

For the converse direction, we prove the following statement by induction on the term s:

$$\forall t \in \mathtt{SP}_\Sigma,\ [\![s]\!] \cong [\![t]\!] \Rightarrow \mathrm{BiMon}_\Box \vdash s = t.$$

$s = 1$: follows from (A.3).
$s = a$: we prove the result by induction on t:
- $t = 1$, $t = [t']$: impossible since $[\![t]\!] \cong \mathbb{o}$;
- $t = b$: since $[\![t]\!] \cong \mathbb{o}$ this means $a = b$, i.e. by reflexivity $\mathrm{BiMon}_\Box \vdash t = a$;
- $t = t_1 \,;\, t_2$: since $\mathbb{o}$ has a single event, and $\mathcal{E}_{[\![t_1 \,;\, t_2]\!]} = \mathcal{E}_{[\![t_1]\!]} \uplus \mathcal{E}_{[\![t_2]\!]}$, that event must be either in $\mathcal{E}_{[\![t_1]\!]}$ or in $\mathcal{E}_{[\![t_2]\!]}$, and the other term has no event. The term that has no event, by (A.3), is provably equal to 1. The term containing an event is isomorphic to $\mathbb{o}$, so by induction it is provably equal to a. Hence we get that t is provably equal to either $a \,;\, 1$ or $1 \,;\, a$, both of which are provably equal to a.
- $t = t_1 \parallel t_2$: same as $t_1 \,;\, t_2$.

$s = s_1 \,;\, s_2$: let $A := \mathcal{E}_{[\![s_1]\!]} \subseteq \mathcal{E}_{[\![s]\!]}$. Notice that $\overline{A} = \mathcal{E}_{[\![s]\!]} \backslash \mathcal{E}_{[\![s_1]\!]} = \mathcal{E}_{[\![s_2]\!]}$. Let ϕ be the isomorphism from $[\![s]\!]$ to $[\![t]\!]$ and let $t_1 := \pi_{\phi(A)}(t)$ and $t_2 := \pi_{\overline{\phi(A)}}(t)$. Since A is nested and prefix, so is its image by the isomorphism ϕ. Therefore by Lemma 2 we get that: $\mathrm{BiMon}_\Box \vdash t = t_1 \,;\, t_2$. By properties of isomorphisms, we can also see that for any set $X \subseteq \mathcal{E}_{[\![s]\!]}$ we have $[\![s]\!]\downarrow_X \cong [\![t]\!]\downarrow_{\phi(X)}$. Hence

we have $[\![t_1]\!] \cong [\![t]\!]\downarrow_{\phi(A)} \cong [\![s]\!]\downarrow_A \cong [\![s_1]\!]$. Similarly, and because by bijectivity of ϕ we have $\overline{\phi(A)} = \phi\left(\overline{A}\right)$, we get $[\![t_2]\!] \cong [\![s_2]\!]$. We may thus conclude by induction that $\mathrm{BiMon}_{\square} \vdash s_i = t_i$ $(i \in \{1,2\})$, i.e.

$$\mathrm{BiMon}_{\square} \vdash s = s_1 \,;\, s_2 = t_1 \,;\, t_2 = t.$$

$s = s_1 \parallel s_2$: same as $s_1 \,;\, s_2$. □

Lemma 4. *Let P, Q be two posets:*

(i) if $P \sqsubseteq_b Q$, then P contains $\boldsymbol{P}_1$ iff Q contains $\boldsymbol{P}_1$;
(ii) if $P \sqsubseteq_o Q$, then P contains $\boldsymbol{P}_2$ iff Q contains $\boldsymbol{P}_2$;
(iii) if $P \sqsubseteq_b Q$ and Q contains $\boldsymbol{P}_3$, then P contains $\boldsymbol{P}_3$;
(iv) if $P \sqsubseteq_b Q$ and Q contains $\boldsymbol{P}_4$, then P contains $\boldsymbol{P}_4$.

Proof. (i) if $\phi : Q \to P$ is order reflecting, then by definition we have

$$x \leq_Q y \Leftrightarrow \phi(x) \leq_P \phi(y).$$

The result follows immediately.

(ii) if $\phi : Q \to P$ is order reflecting, then by definition we have

$$B \in \mathcal{B}_Q \Leftrightarrow \phi\,(B) \in \mathcal{B}_P$$

$$e \in B \setminus C \Leftrightarrow \phi(e) \in \phi\,(B) \setminus \phi\,(C) \qquad e \in B \cap C \Leftrightarrow \phi(e) \in \phi\,(B) \cap \phi\,(C)\,.$$

The result follows immediately.

(iii) if $\phi : Q \to P$ is order reflecting and Q contains $\mathbf{P}_3$ then by definition of the pattern we have $e_1, e_2, e_3 \in \mathcal{E}_Q$ and $B \in \mathcal{B}_Q$ such that

$$e_1 \notin B \qquad e_2 \in B \qquad e_3 \in B \qquad e_1 \leq_Q e_2 \qquad e_1 \not\leq_Q e_3.$$

Since ϕ is a poset homomorphism, we know that $\phi\,(B) \in \mathcal{B}_P$ and $\phi(e_1) \leq_P \phi(e_2)$. By definition of the direct image, we also know that $\phi(e_1) \notin \phi\,(B)$ and $\phi(e_2), \phi(e_3) \in \phi\,(B)$. Finally, since ϕ is order reflecting $\phi(e_1) \not\leq_P \phi(e_3)$.

(iv) similar to the proof for (iii). □

Lemma 7. *There is a function $T_{\_} : \mathtt{T}_{\Sigma} \to \mathcal{P}_f\left(\mathtt{SP}_{\Sigma}\right)$ such that:*

$$\mathrm{SR}_{\square} \vdash e = \sum_{s \in T_e} s. \qquad\qquad [\![e]\!] \cong \{[\![s]\!] \mid s \in T_e\}\,.$$

Proof. T_e is defined by induction on e:

$$\begin{aligned}
T_0 &:= \emptyset & T_1 &:= \{1\} \\
T_a &:= \{a\} & T_{[e]} &:= \{[s] \mid s \in T_e\} \\
T_{e\,;\,f} &:= \{s\,;t \mid \langle s,t\rangle \in T_e \times T_f\} & T_{e \parallel f} &:= \{s \parallel t \mid \langle s,t\rangle \in T_e \times T_f\} \\
T_{e+f} &:= T_e \cup T_f.
\end{aligned}$$

Checking the lemma is done by a simple induction. □

Corollary 1. $[\![e]\!] \cong [\![f]\!] \Leftrightarrow \mathrm{SR}_\Box \vdash e = f$

Proof. Soundness is easy to check: by a simple induction on the derivation tree, we can ensure that $\mathrm{SR}_\Box \vdash e = f \Rightarrow [\![e]\!] \cong [\![f]\!]$.

Using Lemma 7 we may rewrite any term e as a finite union of series parallel terms. Let $s \in T_e$. By soundness, there is $P \in [\![e]\!]$ such that $P \cong [\![s]\!]$. Since $[\![e]\!] \cong [\![f]\!]$, there is $Q \in [\![f]\!]$ such that $P \cong Q$. Since $\mathrm{SR}_\Box \vdash f = \sum_{s \in T_f} s$, by soundness there is $t \in T_f$ such that $Q \cong t$. Therefore $[\![s]\!] \cong [\![t]\!]$, so by Theorem 2 we have $\mathrm{BiMon}_\Box \vdash s = t$. Since $\mathrm{BiMon}_\Box \subseteq \mathrm{SR}_\Box$, we also have $\mathrm{SR}_\Box \vdash s = t$, and because $t \in T_f$ we get $\mathrm{SR}_\Box \vdash s \leq f$. Since this holds for every $s \in T_e$, this means that:

$$\mathrm{SR}_\Box \vdash e = \sum_{s \in T_e} s \leq f.$$

By a symmetric argument we obtain $\mathrm{SR}_\Box \vdash f \leq e$, allowing us to conclude by antisymmetry that $\mathrm{SR}_\Box \vdash e = f$. □

Corollary 3. $[\![e]\!]{\downarrow} \cong [\![f]\!]{\downarrow} \Leftrightarrow \mathrm{CSR}_\Box \vdash e = f$.

Proof. Again, soundness is straightforward. For completeness, it is sufficient to show if $[\![e]\!] \sqsubseteq [\![f]\!]$ then $\mathrm{CSR}_\Box \vdash e \leq f$. Assume $[\![e]\!] \sqsubseteq [\![f]\!]$, i.e. every poset in $[\![e]\!]$ is subsumed by some poset in $[\![f]\!]$. Since $\mathrm{SR}_\Box \subseteq \mathrm{CSR}_\Box$, we get by Lemma 7:

$$\mathrm{CSR}_\Box \vdash e = \sum_{s \in T_e} s \qquad\qquad \mathrm{CSR}_\Box \vdash f = \sum_{t \in T_f} t.$$

Let $s \in T_e$, since $[\![e]\!] \sqsubseteq [\![f]\!]$, and because of Lemma 7, we know that there is a term $t \in T_f$ such that $[\![s]\!] \sqsubseteq [\![t]\!]$. By Theorem 3, that means $\mathrm{CMon}_\Box \vdash s \leq t$. Since $\mathrm{CMon}_\Box \subseteq \mathrm{CSR}_\Box$, we get $\mathrm{CSR}_\Box \vdash s \leq t \leq f$. This means that for every $s \in T_e$, we have $\mathrm{CSR}_\Box \vdash s \leq f$, hence that $\mathrm{CSR}_\Box \vdash e = \sum_{s \in T_e} s \leq f$. □

Lemma 10. *For a formula $\Phi \in \mathtt{F}^+_\Sigma$ and a pair of posets $P, Q \in \mathbb{P}_\Sigma$, if $P \sqsubseteq Q$ then the following hold: $P \models_{J\uparrow} \Phi \Rightarrow Q \models_{J\uparrow} \Phi$, and $Q \models_{J\downarrow} \Phi \Rightarrow P \models_{J\downarrow} \Phi$.*

Proof. We proceed by induction on Φ, first to prove the statement with $J\uparrow$.

- If $\Phi = \bot$, $P \models_{J\uparrow} \bot$ means that $\mathbb{c} \cong P \sqsubseteq Q$, which implies, thanks to Remark 2, that $Q \cong \mathbb{c}$, i.e. $Q \models_{J\uparrow} \bot$.
- If $\Phi = \phi \vee \psi$ or $\Phi = \phi \wedge \psi$, we use the induction hypothesis to show that $P \models_{J\uparrow} \Phi$ implies $Q \models_{J\uparrow} \Phi$.
- If $\Phi = a$, $\Phi = \phi \blacktriangleright \psi$, $\Phi = \phi \star \psi$, $\Phi = [\phi]$, or $\Phi = \langle\!\langle \phi \rangle\!\rangle$, then the satisfaction relation says $P \models_{J\uparrow} \Phi$ iff $P \sqsupseteq P'$ and $h(P')$. Since $Q \sqsupseteq P$, Q also satisfies $Q \sqsupseteq P'$ and $h(P')$ so we can conclude that $Q \models_{J\uparrow} \Phi$ without using the induction hypothesis.

Now we do another induction on Φ to prove the statement with $J\downarrow$.

- If $\Phi = \bot$, $Q \models_{J\downarrow} \bot$ means that $P \sqsubseteq Q \cong \mathbb{c}$, which implies, thanks to Remark 2, that $P \cong \mathbb{c}$, i.e. $P \models_{J\downarrow} \bot$.

– If $\Phi = \phi \vee \psi$ or $\Phi = \phi \wedge \psi$, we use the induction hypothesis to show that $Q \models_{J\downarrow} \Phi$ implies $P \models_{J\downarrow} \Phi$.
– If $\Phi = a$, $\Phi = \phi \blacktriangleright \psi$, $\Phi = \phi \star \psi$, $\Phi = [\phi]$, or $\Phi = (\!|\phi|\!)$, then the satisfaction relation says $Q \models_{J\downarrow} \Phi$ iff $Q \sqsubseteq Q'$ and $h(Q')$. Since $P \sqsubseteq Q$, P also satisfies $P \sqsubseteq Q'$ and $h(Q')$ so we can conclude that $P \models_{J\downarrow} \Phi$ without using the induction hypothesis. □

Lemma 11. *For any sp-term s and any poset P, we have:*

$$P \models_K \phi(s) \Leftrightarrow P \cong [\![s]\!] \qquad P \models_{J\uparrow} \phi(s) \Leftrightarrow P \sqsupseteq [\![s]\!] \qquad P \models_{J\downarrow} \phi(s) \Leftrightarrow P \sqsubseteq [\![s]\!]\,.$$

Proof. By induction on s:

$$\begin{aligned}
s = 1 : \quad & P \models_K \phi(1) = \bot \Leftrightarrow P \cong \mathbb{c} = [\![1]\!]\,.\\
& P \models_{J\uparrow} \phi(1) = \bot \Leftrightarrow P \cong \mathbb{c} \Leftrightarrow P \sqsupseteq \mathbb{c} = [\![1]\!]\,.\\
& P \models_{J\downarrow} \phi(1) = \bot \Leftrightarrow P \cong \mathbb{c} \Leftrightarrow P \sqsubseteq \mathbb{c} = [\![1]\!]\,.\\
s = a : \quad & P \models_K \phi(a) = a \Leftrightarrow P \cong \mathbb{a} = [\![a]\!]\,.\\
& P \models_{J\uparrow} \phi(a) = a \Leftrightarrow P \sqsupseteq \mathbb{a} = [\![a]\!]\,.\\
& P \models_{J\downarrow} \phi(a) = a \Leftrightarrow P \sqsubseteq \mathbb{a} = [\![a]\!]\,.\\
s = [t] : \quad & P \models_K [\phi(t)] \Leftrightarrow P \cong [Q] \wedge Q \models_K \phi(t)\\
& \qquad\qquad \Leftrightarrow P \cong [Q] \wedge Q \cong [\![t]\!]\\
& \qquad\qquad \Leftrightarrow P \cong [[\![t]\!]] = [\![[t]]\!]\,.\\
& P \models_{J\uparrow} [\phi(t)] \Leftrightarrow P \sqsupseteq [Q] \wedge Q \models_{J\uparrow} \phi(t)\\
& \qquad\qquad \Leftrightarrow P \sqsupseteq [Q] \wedge Q \sqsupseteq [\![t]\!]\\
& \qquad\qquad \Leftrightarrow P \sqsupseteq [[\![t]\!]] = [\![[t]]\!]\,.\\
& P \models_{J\downarrow} [\phi(t)] \Leftrightarrow P \sqsubseteq [Q] \wedge Q \models_{J\downarrow} \phi(t)\\
& \qquad\qquad \Leftrightarrow P \sqsubseteq [Q] \wedge Q \sqsubseteq [\![t]\!]\\
& \qquad\qquad \Leftrightarrow P \sqsubseteq [[\![t]\!]] = [\![[t]]\!]\,.\\
s = s_1\,;s_2 : \quad & P \models_K \phi(s_1) \blacktriangleright \phi(s_2) \Leftrightarrow P \cong P_1 \otimes P_2\\
& \qquad\qquad \wedge P_1 \models_K \phi(s_1) \wedge P_2 \models_K \phi(s_2)\\
& \qquad\qquad \Leftrightarrow P \cong P_1 \otimes P_2 \wedge P_1 \cong [\![s_1]\!] \wedge P_2 \cong [\![s_2]\!]\\
& \qquad\qquad \Leftrightarrow P \cong [\![s_1]\!] \otimes [\![s_2]\!] = [\![s_1\,;s_2]\!]\,.\\
& P \models_{J\uparrow} \phi(s_1) \blacktriangleright \phi(s_2) \Leftrightarrow P \sqsupseteq P_1 \otimes P_2\\
& \qquad\qquad \wedge P_1 \models_{J\uparrow} \phi(s_1) \wedge P_2 \models_{J\uparrow} \phi(s_2)\\
& \qquad\qquad \Leftrightarrow P \sqsupseteq P_1 \otimes P_2 \wedge P_1 \sqsupseteq [\![s_1]\!] \wedge P_2 \sqsupseteq [\![s_2]\!]\\
& \qquad\qquad \Leftrightarrow P \sqsupseteq [\![s_1]\!] \otimes [\![s_2]\!] = [\![s_1\,;s_2]\!]\,.\\
& P \models_{J\downarrow} \phi(s_1) \blacktriangleright \phi(s_2) \Leftrightarrow P \sqsubseteq P_1 \otimes P_2\\
& \qquad\qquad \wedge P_1 \models_{J\downarrow} \phi(s_1) \wedge P_2 \models_{J\downarrow} \phi(s_2)\\
& \qquad\qquad \Leftrightarrow P \sqsubseteq P_1 \otimes P_2 \wedge P_1 \sqsubseteq [\![s_1]\!] \wedge P_2 \sqsubseteq [\![s_2]\!]\\
& \qquad\qquad \Leftrightarrow P \sqsubseteq [\![s_1]\!] \otimes [\![s_2]\!] = [\![s_1\,;s_2]\!]\,.
\end{aligned}$$

$$
\begin{aligned}
s = s_1 \parallel s_2 : \quad P \models_K \phi(s_1) \star \phi(s_2) &\Leftrightarrow P \cong P_1 \oplus P_2 \\
&\quad \wedge P_1 \models_K \phi(s_1) \wedge P_2 \models_K \phi(s_2) \\
&\Leftrightarrow P \cong P_1 \oplus P_2 \wedge P_1 \cong [\![s_1]\!] \wedge P_2 \cong [\![s_2]\!] \\
&\Leftrightarrow P \cong [\![s_1]\!] \oplus [\![s_2]\!] = [\![s_1 \parallel s_2]\!] . \\
P \models_{J\uparrow} \phi(s_1) \star \phi(s_2) &\Leftrightarrow P \sqsupseteq P_1 \oplus P_2 \\
&\quad \wedge P_1 \models_{J\uparrow} \phi(s_1) \wedge P_2 \models_{J\uparrow} \phi(s_2) \\
&\Leftrightarrow P \sqsupseteq P_1 \oplus P_2 \wedge P_1 \sqsupseteq [\![s_1]\!] \wedge P_2 \sqsupseteq [\![s_2]\!] \\
&\Leftrightarrow P \sqsupseteq [\![s_1]\!] \oplus [\![s_2]\!] = [\![s_1 \parallel s_2]\!] . \\
P \models_{J\downarrow} \phi(s_1) \star \phi(s_2) &\Leftrightarrow P \sqsubseteq P_1 \oplus P_2 \\
&\quad \wedge P_1 \models_{J\downarrow} \phi(s_1) \wedge P_2 \models_{J\downarrow} \phi(s_2) \\
&\Leftrightarrow P \sqsubseteq P_1 \oplus P_2 \wedge P_1 \sqsubseteq [\![s_1]\!] \wedge P_2 \sqsubseteq [\![s_2]\!] \\
&\Leftrightarrow P \sqsubseteq [\![s_1]\!] \oplus [\![s_2]\!] = [\![s_1 \parallel s_2]\!] .
\end{aligned}
$$

□

Lemma 12. *The following statements hold universally:*

$$e \models^{\forall}_K \phi \Leftrightarrow [\![e]\!] \subsetneq [\![\phi]\!]_K \tag{3.2}$$

$$e \models^{\forall}_{J\downarrow} \phi \Leftrightarrow [\![e]\!] \sqsubseteq [\![\phi]\!]_{J\downarrow} \tag{3.3}$$

$$e \models^{\forall}_{J\uparrow} \phi \Leftrightarrow \forall P \in [\![e]\!] , \exists Q \in [\![\phi]\!]_{J\uparrow} : P \sqsupseteq Q. \tag{3.4}$$

$$e \subsetneq f \Rightarrow \forall \phi, e \models^{\exists}_K \phi \Rightarrow f \models^{\exists}_K \phi \tag{3.5}$$

$$e \sqsubseteq f \Rightarrow \forall \phi, e \models^{\exists}_{J\uparrow} \phi \Rightarrow f \models^{\exists}_{J\uparrow} \phi \tag{3.6}$$

$$e \subsetneq f \Rightarrow \forall \phi, f \models^{\forall}_K \phi \Rightarrow e \models^{\forall}_K \phi \tag{3.7}$$

$$e \sqsubseteq f \Rightarrow \forall \phi, f \models^{\forall}_{J\downarrow} \phi \Rightarrow e \models^{\forall}_{J\downarrow} \phi \tag{3.8}$$

$$e \models^{\exists}_K \phi(s) \Leftrightarrow [\![s]\!] \in [\![e]\!] \tag{3.9}$$

$$e \models^{\exists}_{J\uparrow} \phi(s) \Leftrightarrow [\![s]\!] \in [\![e]\!]\downarrow \tag{3.10}$$

Proof. We prove each statement in sequence:

(3.2),(3.4),(3.3) By (3.1) and Lemmas 9 and 10.

(3.5) if $e \models^{\exists}_K \phi$ and $e \subsetneq f$, then $f \models^{\exists}_K \phi$:
The first premise the existence of a poset $P \in [\![e]\!] \cap [\![\phi]\!]_K$. The second allows us to find a poset $Q \in [\![f]\!]$ such that $P \cong Q$. Since $P \in [\![\phi]\!]_K$ and $[\![\phi]\!]_K$ is closed under $\cong$ we get $Q \in [\![f]\!] \cap [\![\phi]\!]_K$.

(3.6) if $e \models^{\exists}_{J\uparrow} \phi$ and $e \sqsubseteq f$, then $f \models^{\exists}_{J\uparrow} \phi$:
The first premise the existence of a poset $P \in [\![e]\!] \cap [\![\phi]\!]_{J\uparrow}$. The second allows us to find a poset $Q \in [\![f]\!]$ such that $P \sqsubseteq Q$. Since $P \in [\![\phi]\!]_{J\uparrow}$ and $[\![\phi]\!]_{J\uparrow}$ is upwards-closed we get $Q \in [\![f]\!] \cap [\![\phi]\!]_{J\uparrow}$.

(3.7) if $f \models^{\forall}_K \phi$ and $e \subsetneq f$, then $e \models^{\forall}_K \phi$:
The premise tells us that $[\![f]\!] \subsetneq [\![\phi]\!]_K$ and $[\![e]\!] \subsetneq f$, so by transitivity we get that $[\![e]\!] \subsetneq [\![\phi]\!]_K$, i.e. $e \models^{\forall}_K \phi$.

(3.8) if $f \models^{\forall}_{J\downarrow} \phi$ and $e \sqsubseteq f$, then $e \models^{\forall}_{J\downarrow} \phi$:
The premise tells us that $[\![f]\!] \sqsubseteq [\![\phi]\!]_{J\downarrow}$ and $[\![e]\!] \sqsubseteq f$, so by transitivity we get that $[\![e]\!] \sqsubseteq [\![\phi]\!]_{J\downarrow}$, i.e. $e \models^{\forall}_{J\downarrow} \phi$.
(3.9) $e \models^{\exists}_{K} \phi(s) \Leftrightarrow [\![e]\!] \cap [\![\phi(s)]\!]_K \neq \emptyset \Leftrightarrow [\![e]\!] \cap \{[\![s]\!]\} \neq \emptyset \Leftrightarrow [\![s]\!] \in [\![e]\!]$.
(3.10) $e \models^{\exists}_{J\uparrow} \phi(s) \Leftrightarrow [\![e]\!] \cap [\![\phi(s)]\!]_{J\uparrow} \neq \emptyset \Leftrightarrow [\![e]\!] \cap \{[\![s]\!]\}\uparrow \neq \emptyset \Leftrightarrow [\![s]\!] \in [\![e]\!]\downarrow$. □

Proposition 1. *For a pair of series parallel terms s,t, the identity* $\mathrm{BiMon}_\Box \vdash s = t$ *is derivable if and only if for any formula ϕ, $[\![s]\!] \models_K \phi$ iff $[\![t]\!] \models_K \phi$.*

Proof.

$$
\begin{aligned}
\mathrm{BiMon}_\Box \vdash s = t &\Leftrightarrow [\![s]\!] \cong [\![t]\!] && \text{(Theorem 2)}\\
&\Rightarrow \forall \phi \in \mathtt{F}_\Sigma,\ [\![s]\!] \models_K \phi \Leftrightarrow [\![t]\!] \models_K \phi. && \text{(Lemma 9)}\\
&\Rightarrow [\![s]\!] \models_K \phi(t) \quad \text{(Since } [\![t]\!] \models_K \phi(t) \text{ follows from Lemma 11)}\\
&\Leftrightarrow [\![s]\!] \cong [\![t]\!] && \text{(Lemma 11)}\\
&\Leftrightarrow \mathrm{BiMon}_\Box \vdash s = t && \text{(Theorem 2)}
\end{aligned}
$$

□

Proposition 2. *For a pair of terms $s,t \in \mathtt{SP}_\Sigma$, the identity* $\mathrm{CMon}_\Box \vdash s \leq t$ *is derivable if and only if for any formula ϕ, $[\![s]\!] \models_{J\uparrow} \phi$ implies $[\![t]\!] \models_{J\uparrow} \phi$.*

Proof.

$$
\begin{aligned}
\mathrm{CMon}_\Box \vdash s \leq t &\Leftrightarrow [\![s]\!] \sqsubseteq [\![t]\!] && \text{(Theorem 3)}\\
&\Rightarrow \forall \phi \in \mathtt{F}^{+}_{\Sigma},\ [\![s]\!] \models_{J\uparrow} \phi \Rightarrow [\![t]\!] \models_{J\uparrow} \phi. && \text{(Lemma 10)}\\
&\Rightarrow [\![t]\!] \models_{J\uparrow} \phi(s) \quad \text{(Since } [\![s]\!] \models_{J\uparrow} \phi(s) \text{ follows from Lemma 11)}\\
&\Leftrightarrow [\![t]\!] \sqsupseteq [\![s]\!] && \text{(Lemma 11)}\\
&\Leftrightarrow \mathrm{CMon}_\Box \vdash s \leq t. && \text{(Theorem 3)}
\end{aligned}
$$

□