Bringing order to the separation logic jungle

Qinxiang Cao, Santiago Cuellar, and Andrew W. Appel

Princeton University

Abstract. Research results from so-called "classical" separation logics are not easily ported to so-called "intuitionistic" separation logics, and vice versa. Basic questions like, "Can the frame rule be proved independently of whether the programming language is garbage-collected?" "Can amortized resource analysis be ported from one separation logic to another?" should be straightforward. But they are not. Proofs done in a particular separation logic are difficult to generalize. We argue that this limitation is caused by incompatible semantics. For example, emp sometimes holds everywhere and sometimes only on units.
In this paper, we introduce a unifying semantics and build a framework that allows to reason parametrically over all separation logics. Many separation algebras in the literature are accompanied, explicitly or implicitly, by a preorder. Our key insight is to axiomatize the interaction between the join relation and the preorder. We prove every separation logic to be sound and complete with respect to this unifying semantics. Further, our framework enables us to generalize the soundness proofs for the frame rule and CSL. It also reveals a new world of meaningful intermediate separation logics between "intuitionistic" and "classical".

Keywords: Separation logic, Order, Separation Algebra

1 Introduction

Separation logic, introduced at the turn of the millennium by Reynolds [32], has led to tremendous progress in modular verification in a multitude of applications: for reasoning about multiple languages, including those with C-like (malloc/free) or Java-like (garbage-collected) memory model; for reasoning about concurrency; for amortized resource analysis; or for static automated bug analysis. Unfortunately, many of those instances disagree on the definitions of separation logic itself. In 2010, Parkinson [27] warned us about this indiscriminate proliferation of separation logics.

> The problem with this proliferation of logics is that each logic requires a new soundness proof.
>
> — *The Next 700 Separation Logics*

Seven years later, separation logics are still tailored for particular uses, which makes them incompatible and hard to extend; they have different proof rules, they use custom underlying models and impose conflicting semantics.

To appear in *APLAS'17: 15th Asian Symposium on Programming Languages and Systems,* November 2017.

The best example of the divide originates from the conventional wisdom [1][5][18][28] that "intuitionistic separation logics are for garbage-collected languages, classical separation logics are for malloc/free languages." We think that intuitionistic vs. classical is not the right way to look at it; the problem with the nomenclature is twofold.

First, it creates a false dichotomy. The literature has almost exclusively dealt with these two flavors of logic, while ignoring a large variety of intermediate logics. For example, it is known that the elimination rule of separating conjunction ($\vdash \varphi * \psi \to \varphi$) is incompatible with the law of excluded middle [19] but, as we show in this paper, it is compatible with some weaker forms. The resulting intermediate separation logics are rich and meaningful but have largely been neglected. In fact, we found separation logic instances that admitted intermediate rules unbeknownst to the authors [1,17,11].

Second, the naming convention creates two seemingly incompatible bodies of work. Since authors choose one or the other framework, results are often hard to extend and different works are hard to compare. For instance, the soundness of the *frame rule* given the *frame property* [19] has to be proved anew for every different semantic model. Similarly, the soundness of concurrent separation logic proved by Brookes [7] does not naturally extend to garbage-collected languages. Other works that are hard to extend include the discussion of *preciseness* and the conjunction rule [26] [16], and recent advances in logics for concurrency [22].

The main difficulty in unifying the two worlds is that authors use different and sometimes incompatible semantics. For instance, one side enforces that emp holds only on units, while for the other side it is simply equal to True. In fact, even within each side there are several conflicting semantics.

Ideally, the semantics of the separating operators would be given by the intuitive definitions

$$m \vDash \varphi * \psi \triangleq \text{exists } m_1\ m_2 \text{ s.t. } \oplus(m_1, m_2, m) \text{ and } m_1 \vDash \varphi \text{ and } m_2 \vDash \psi \quad (4.1)$$

$$m \vDash \varphi -\!\!* \psi \triangleq \text{for any } m_1\ m_2, \text{ if } \oplus(m, m_1, m_2), m_1 \vDash \varphi \text{ implies } m_2 \vDash \psi \quad (4.2)$$

where $\oplus$ is a join operation on the underlying model, called a *separation algebra* [10]. Unfortunately, when the objects of interest are not just simple heaplets—when step-indexing [17][6] or topology [31, §3] or amortized analysis [4] or buffer flushing [11] is involved—authors have had to define more intricate semantics; the simple semantics is (apparently) unsound. We explain this in section 5.3. Not only are these "messier" semantic definitions inconvenient, they cause non-portability of results.

In this paper we show that all of those semantics are in fact instances of a *flat semantics* over the generalized *ordered separation algebras*. An ordered separation algebra is just a separation algebra together with a preorder $\leq$.

Ordered separation algebras are not a new idea. In fact, the heap model defined by Reynolds in his first paper on separation logic [32] is ordered by heap extension. Similarly, the monotonic state of Pilkiewicz and Pottier [29] and the amortized resource of Atkey [4] are also ordered. Furthermore, Pym *et al.* [31], Galmiche *et al.* [15] and Jensen [20] have used ordered separation algebras

explicitly as their semantic model. Despite this common trend, orders have been used to define different semantics, tailored to specific models, and no unification had yet been discovered.

Contributions. We argue that all models of separation logic are ordered separation algebras (section 5.1) and we show that all semantics in the literature, to the best of our knowledge, can be formulated as instances of our *flat semantics* (thm. 3). Our unification holds for the "classical" side, the "intuitionistic" side and all separation logics in between.

By this unifying semantics, we establish a correspondence between all different separation logics and classes of ordered separation algebras. We prove that any separation logic is sound and complete w.r.t. flat semantics in its corresponding class of models (section 6).

We generalize two theoretical applications of separation logics (section 7). We show that the frame rule (given frame property) and CSL are sound parametrically on different separation logic semantics.

All the definitions, propositions, lemmas and theorems in this paper have been formalized in Coq. We will often omit uninteresting proofs, but a curious reader can find them in our publicly available development.

2 Related Work

Authors use both "separation logic" and "bunched logic" to describe a propositional logic extended with separating connectives ($*$ and $-\!\!*$). Usually, "bunched logics" (or bunched implication, BI) [25] are used when the semantics are defined over abstract relational Kripke models while"separation logics" [32] are used when the semantics are defined over some specific memory model. This distinction is not absolute. Calcagno *el. al* [10] first studied separation logic over abstract models, i.e. separation algebras. In this paper, we always use the name "separation logic". What the bunched logic literatures call BI is, in our terminology, the proof theory IP + SL; what they call Boolean BI (BBI) we characterize as IP + SL + EM. Some authors also use "separation logic" to describe Hoare logics whose assertion languages contain $*$ and $-\!\!*$. In this paper, we call them Hoare separation logics.

Ishtiaq and O'Hearn [19] showed a modal translation $(\cdot)^\circ$ that embeds separation logic without excluded middle [32] into their logic with excluded middle. The translation preserves validity; i.e $s, h \vDash^{Reynolds} P \Leftrightarrow s, h \vDash^{Ishtiaq} P^\circ$. The translation is enough to show the soundness of both logics with respect to their model, but not for other models. Of particular interest, they show the *frame rule* is sound w.r.t. operational semantics with the *frame property*. We extend the soundness of the frame rule, parametrically on different semantic models.

Galmiche and Larchey-Wendling [14] prove the completeness of classical separation logics (BBI). Our soundness and completeness proof generalizes their result to nonclassical logics.

Pym, O'Hearn and Yang [31] derive a Kripke semantic model for separation logics (called PDM) from topology semantics and prove its soundness and

completeness. It is later used by other authors as a unifying semantic model. This semantics is equivalent to downwards semantics in this paper. Their model cannot cover step indexed models [17][6] and some variants of FSCQ's semantic model [11], which are instances of upwards semantics in this paper.

Jensen [20] has a thorough review of separation logics and separation algebras. The chapter is expositional, but he explicitly imposes an order on each separation algebra. Our presentation of separation logic semantics is closely related to his. In particular, his Propositions 3.6 and 3.7 correspond to our definitions 13.1 (upwards semantics) and 13.2 (downwards semantics), of which he writes:

> The conditions of neither Proposition 3.6 nor Proposition 3.7 generalise the conditions of the other, so perhaps a unifying theorem is still waiting to be discovered.

Our theorem 3 is exactly that theorem.

It is well known that separation logic is a kind of binary modal logic [31]. Some authors have proposed different semantics of intuitionistic (unary) modal logic—Simpson [33] showed that the main disagreement among them is a choice between requiring the frame to satisfy a closure property and building the monotonicity property into the semantic definition of modalities. Our discussion of upwards semantics, downwards semantics and flat semantics for separation logic is similar to that discussion in Simpson's thesis about modal logic. But we go further in this paper, we show a unifying solution of all these semantics.

3 Taxonomy Of Separation Logic

In this section, we formally define the scope of separation logics based on their proof systems. In turn, this allows us to group the separation logics to provide parametric proofs of soundness and completeness in section 6. This classification will hopefully dispel the unfortunate nomenclature (classical vs. intuitionistic) that has been prevalent.

3.1 Defining separation logic

We take the syntax below to be the assertion language of separation logic.

Definition 1 (Separation logic syntax). *For a given set of atomic assertions Σ, we use $\mathcal{L}(\Sigma)$ to represent the smallest language with all following assertions:*

$$\varphi ::= p(\in \Sigma) \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \to \varphi_2 \mid \varphi_1 * \varphi_2 \mid \varphi_1 \mathrel{-\!\!*} \varphi_2 \mid \bot \mid \mathsf{emp}$$

As usual, we use the following connectives as abbreviations: $\varphi_1 \leftrightarrow \varphi_2 \triangleq (\varphi_1 \to \varphi_2) \wedge (\varphi_2 \to \varphi_1)$, $\neg\varphi \triangleq \varphi \to \bot$, $\top \triangleq \bot \to \bot$.

We will define separation logics based on a Hilbert-style proof system; that is a set of axioms and proof rules. Intuitionistic propositional logic is defined by the the rules IP in fig. 1. EM, GD and WEM in fig. 2 are optional axioms for

$$\frac{\vdash \varphi \qquad \vdash \varphi \to \psi}{\vdash \psi}\ (\text{MP}) \qquad \vdash \varphi \to \psi \to \varphi \wedge \psi\ (\wedge\text{I}) \qquad \vdash \varphi \wedge \psi \to \psi\ (\wedge\text{E}_2) \qquad \vdash \psi \to \varphi \vee \psi\ (\vee\text{I}_2)$$

$$\vdash \varphi \to (\psi \to \varphi)\ (\to_1) \qquad \vdash \varphi \wedge \psi \to \varphi\ (\wedge\text{E}_1) \qquad \vdash \varphi \to \varphi \vee \psi\ (\vee\text{I}_1) \qquad \vdash \bot \to \varphi\ (\bot\text{E})$$

$$\vdash (\varphi \to \psi \to \chi) \to (\varphi \to \psi) \to (\varphi \to \chi)\ (\to_2) \qquad \vdash (\varphi \to \chi) \to (\psi \to \chi) \to (\varphi \vee \psi \to \chi)\ (\vee\text{E})$$

Fig. 1: Axioms and rules of intuitionistic propostional logic (IP)

$$\vdash \neg\varphi \vee \neg\neg\varphi \quad (\text{WEM}) \qquad \vdash \varphi \to \psi \vee \psi \to \varphi \quad (\text{GD}) \qquad \vdash \varphi \vee \neg\varphi \quad (\text{EM})$$

Fig. 2: Optional axioms for propositional logic

$$\vdash \varphi * \psi \to \varphi \quad (*\text{E}) \qquad \vdash \mathsf{emp} \wedge (\varphi * \psi) \ \to\ \varphi \quad (\text{eE}) \qquad \vdash \mathsf{emp} \wedge \varphi \ \to\ \varphi * \varphi \ (\text{eDUP})$$

Fig. 3: Optional axioms for separating connectives

propositional logic. A logic with EM is classical, while the weaker axioms GD and WEM give rise to intermediate logics. Many other similar axioms give rise to more intermediate logics; we omit them here for the sake of space.

The axioms and rules for minimun separation logic (SL) are in fig. 4: commutativity, associativity, adjoint property of the separating operators, monotonicity of separation over implication and the fact the emp is a separation unit. The double line in $*$ADJ implies the derivation works both ways. The axioms $*$E, eE and eDUP in fig. 3 are optional: $*$E is the elimination rule of the separating conjunction; eE enforces that every empty piece to be non-splittable; and eDUP says all empty pieces are duplicable.

$$\frac{}{\vdash \varphi * \psi \to \psi * \varphi}\ (*\text{COMM}) \qquad \frac{\vdash \varphi * \psi \to \chi}{\vdash \varphi \to \psi \mathrel{-\!\!*} \chi}\ (*\text{ADJ}) \qquad \frac{}{\vdash \varphi * \mathsf{emp} \leftrightarrow \varphi}\ (\text{EMP})$$

$$\frac{}{\vdash (\varphi * \psi) * \chi \to \varphi * (\psi * \chi)}\ (*\text{ASSOC}) \qquad \frac{\vdash \varphi_1 \to \psi_1 \qquad \vdash \varphi_2 \to \psi_2}{\vdash \varphi_1 * \varphi_2 \to \psi_1 * \psi_2}\ (*\text{MONO})$$

Fig. 4: Axioms and rules for separating connectives (SL)

Even with the axioms and proof rules in minimum separation logic (IP + SL), many useful properties can be derived. For example, all of the following assertions are provable in any separation logic: $\vdash (\varphi \mathrel{-\!\!*} \psi) * \varphi \to \psi$, $\vdash (\varphi \vee \psi) * \chi \leftrightarrow (\varphi * \chi \vee \psi * \chi)$, $\vdash (\varphi \wedge \psi) * \chi \to (\varphi * \chi \wedge \psi * \chi)$.

To dispel the unfortunate nomenclature, we call a separation logic with EM *classical*, those with $*$E *garbage-collected* and those with eE and eDUP *malloc/free*.

Degenerate separation logic Reynolds [32] and Ishtiaq and O'Hearn [19] informally postulated that EM and $*$E are "incompatible". It turns out that if a separation logic Γ is classical and garbage-collected, its separating connectives collapses, i.e. for any φ and ψ: $\vdash^{\Gamma} \varphi * \psi \leftrightarrow \varphi \wedge \psi$ and $\vdash^{\Gamma} (\varphi \mathrel{-\!\!*} \psi) \leftrightarrow (\varphi \to \psi)$.

Brotherston and Kanovich [8] prove that a separation logic collapses if it is malloc/free and garbage-collected at the same time. They also show that a classical and garbage-collected separation logic is malloc/free.

4 Background

4.1 Separation Algebra

The semantics of all separation logics are built on structures (e.g., heaps, histories, amortized resources) that share common properties such as associativity and commutativity.

Definition 2 (Commutativity). *For a set M, the relation $\oplus \subseteq M \times M \times M$ is* commutative *iff for all $m_1,\ m_2,$ and m: $\oplus(m_1, m_2, m)$ implies $\oplus(m_2, m_1, m)$.*

Definition 3 (Associativity). *For a set M, the relation $\oplus \subseteq M \times M \times M$ is* associative *iff for all $m_x\ m_y\ m_z\ m_{xy}$ and m_{xyz}, if $\oplus(m_x, m_y, m_{xy})$ and $\oplus(m_{xy}, m_z, m_{xyz})$, there exists m_{yz} such that $\oplus(m_y, m_z, m_{yz})$ and $\oplus(m_x, m_{yz}, m_{xyz})$*

Calcagno, O'Hearn and Yang [10] first called such structures $(M, \oplus)$ *separation algebras* and proposed them to be also cancellative, functional (i.e. $\oplus(a, b, c)$ and $\oplus(a, b, c')$ implies $c = c'$) and with a unit. Since then, the definition has been revisited several times [13] [16] [12] [30]. We follow Brotherston and Villard [9] in that the $\oplus$ relation is neither functional nor cancellative. We depart from them in that we require no units, so we only require a separation algebra to be commutative and associative.

Using the algebras, Calcagno, O'Hearn and Yang define the semantics of separation logic as in definition 4. This is the most widely used and intuitive definition. Foreshadowing the stronger definition (12), we will call them *weak semantics*.

Definition 4 (Weak Semantics).

$$m \vDash \varphi * \psi \triangleq \textit{exists } m_1\ m_2 \textit{ s.t.} \oplus (m_1, m_2, m) \textit{ and } m_1 \vDash \varphi \textit{ and } m_2 \vDash \psi \tag{4.1}$$

$$m \vDash \varphi \mathrel{-\!\!*} \psi \triangleq \textit{for any } m_1\ m_2\textit{, if } \oplus (m, m_1, m_2),\ m_1 \vDash \varphi \textit{ then } m_2 \vDash \psi \tag{4.2}$$

4.2 Kripke semantics for intuitionistic logic

The Kripke semantics [23] for the propositional language is defined on Kripke models.

Definition 5 (Kripke model). *For an intuitionistic logic with atomic proposition set Σ, a* Kripke model *is a tuple $(M, \leq, J)$ in which*

1. $\leq$ *is a preorder on* M
2. $J \in M \to \mathcal{P}(\Sigma)$ *is an interpretations of atomic propositions and is monotonic, i.e., for any* $m, n \in M$*, if* $m \leq n$ *then* $J(m) \subseteq J(n)$*.*

We also call such tuple $(M, \leq)$ *a* Kripke structure*.*

Definition 6 (Kripke Semantics). *Given a Kripke model* $(M, \leq, J)$*, the satisfaction relation* $(\cdot \vDash_{(M,\leq,J)} \cdot)$ *is defined as follows (when the Kripke model is unambiguous from the context, we will omit it for conciseness)*

$m \vDash p \triangleq\ p \in J(m)$ $\qquad$ $m \vDash \varphi \vee \psi \triangleq\ m \vDash \varphi$ *or* $m \vDash \psi$

$m \vDash \bot \triangleq$ *never* $\qquad$ $m \vDash \varphi \to \psi \triangleq$ *for any* m_0*, if* $m \leq m_0$ *then* $m_0 \vDash \varphi$ *implies* $m_0 \vDash \psi$

$m \vDash \varphi \wedge \psi \triangleq\ m \vDash \varphi$ *and* $m \vDash \psi$

Intuitionistic propositional logic is sound and complete w.r.t. this semantics. Moreover, Kripke semantics is a unifying solution of intuitionistic and classical propositional logics: the identity relation is a trivial order and classical logic is sound and complete w.r.t. Kripke semantics in all models with a discrete order.

5 Model And Semantics

In this section, we introduce a framework to define *flat semantics* on *ordered separation algebras* which unifies all different semantics of separation logic (we know of) in the literature.

We introduce *ordered separation algebras*, define properties that they and their elements may have (*increasing*, *unital*, *upwards-closed*, *downwards-closed*). We show examples from the literature of algebras with and without these properties. We show ways of constructing downwards-closed upwards-closed algebras. We demonstrate examples from the literature of different semantics of separation logic and we show that they are all equivalent to instances of *flat semantics*.

5.1 Ordered separation algebras

Using a Kripke semantics is a motivation to impose an order over separation algebras and in fact this is a very common practice (implicitly or explicitly). For example, the heap model defined by Reynolds in his first paper of separation logic [32] is ordered by heap extension, and the resources of a monotonically increasing counter [29] [21] are ordered by the order of natural numbers.

More interesting examples are those that impose an execution order. For instance, states that are step indexed [2] will be ordered by the index, which decreases with execution. In the same vein, Pottier [30] has an *active execution order*[1] for elements of his algebra. The separation algebra by Jung *et al.* [6] is ordered both by heap extension and step index.

[1] Pottier also adds a *passive execution order* which constitutes what he calls a *monotonic separation algebra*. The idea is similar but goes in a different direction, aiming for a type system and not a separation logic.

Since the identity relation is also a preorder, it is not overly restrictive to require separation algebras to be ordered—any separation algebra is trivially ordered by the discrete order. With that in mind we define *ordered separation algebra* as a more expressive way to view separation algebras.

Definition 7 (Ordered separation algebra). *An ordered separation algebra is a tuple $(M, \leq, \oplus)$, where M is the carrier set, $\leq$ is a preorder on M and $\oplus \subseteq M \times M \times M$ is a three-place relation that is commutative and associative.*

Just like in Dockins et al. [13], OSA form an algebra closed under cartesian product, disjoint sum and exponentiation.

If the order is irrelevant or can be inferred by context we will simply call it a *separation algebra*. Also, we call a tuple $(M, \leq, \oplus, J)$ an extended Kripke model if $(M, \leq, \oplus)$ is an OSA and $(M, \leq, J)$ is a Kripke model.

The most obvious example of OSAs are heaps $\mathcal{H}_V : \mathsf{adr} \rightharpoonup V$, which are most commonly discrete (i.e. ordered by $\Delta_2 \triangleq \{(h, h) \mid h \in \mathcal{H}_V\}$) or ordered by heap extension (i.e. $\sqsubseteq_{\mathcal{H}_V} \triangleq \{(h_1, h_2) \mid \mathsf{dom}\ h_1 \subseteq \mathsf{dom}\ h_2 \quad \text{and} \quad \forall x \in \mathsf{dom}\ h_1, h_1(x) = h_2(x)\}$). Many other resources are OSAs, such as ordered monoids, which are used for amortized resource analysis [4].

5.2 Increasing elements and algebras

Definition 7 imposes almost no constraints on $\leq$ other than being a preorder. Particularly, there is no relation between $\oplus$ and $\leq$. We do this in order to be as general as possible and we will revisit this goal in section 5.3. However, it is worth spelling out an intuitive and important connection between the two relations. In heaps with heap-extension order, joining two heaps always results in a larger heap. We capture this property in the following definitions.

Definition 8 (Increasing element and algebra). *For a separation algebra $(M, \leq, \oplus)$, an* increasing *element $e \in M$ is one such that for all $m, n \in M$, if $\oplus(e, m, n)$ then $m \leq n$. Similarly, a separation algebra is increasing if all its elements are increasing.*

The increasing OSAs also form an algebra closed under cartesian product, disjoint sum and exponentiation. The intuition towards *increasing* will be made clear throughout the paper; it is used for the semantics of emp (section 5.5) and it is key for defining separation logics for garbage-collected languages (thm.5).

5.3 Extending Kripke semantics

It is a fundamental property of Kripke semantics that the denotations of all assertions are monotonic, not just those of atomic assertions. More precisely,

Definition 9 (Monotonic denotation). *An assertion φ has monotonic denotation in a Kripke model $(M, \leq)$ w.r.t. a semantics $(\cdot \vDash \cdot)$ iff for any $n, m \in M$,*

$$\textit{If } n \leq m \textit{ then } n \vDash \varphi \textit{ implies } m \vDash \varphi \tag{1}$$

This property is critical in proving soundness of intuitionistic logic and any of its extensions (e.g. intuitionistic first order logic or intuitionistic modal logic).

Therefore, in extending Kripke semantics with separating connectives, property (1) must be preserved. To do so, if separating conjunction is defined as in (4.1), then the separation algebra must be *upwards-closed*:

Definition 10 (Upwards-closed separation algebra (UCSA)). *An OSA* $(M, \leq, \oplus)$ *is* upwards-closed *if the join relation is upwards-closed with respect to the order. In other words, for all* m_1, m_2, m *and* n, *If* $\oplus(m_1, m_2, m)$ *and* $m \leq n$, *then there exist* n_1 *and* n_2 *s.t.* $\oplus(n_1, n_2, n)$*and* $m_1 \leq n_1$ *and* $m_2 \leq n_2$.

Similarly, if separating implication is defined as in (4.2), then we must require our separation algebra to be *downwards-closed*:

Definition 11 (Downwards-closed separation algebra (DCSA)). *An OSA* $(M, \leq, \oplus)$ *is* downwards-closed *if the join relation is downwards-closed w.r.t. the order. In other words, for all* m_1, m_2, m, n_1 *and* n_2 *: If* $\oplus\,(m_1, m_2, m), n_1 \leq m_1$ *and* $n_2 \leq m_2$ *then there exists* n *s.t.* $\oplus\,(n_1, n_2, n)$ *and* $n \leq m$.

Theorem 1 (Weak semantics monotonicity). *For any assertions* φ *and* ψ *with monotonic denotation,*

1. *the weak denotation of* $\varphi * \psi$ (4.1) *is monotonic in any UCSA*
2. *the weak denotation of* $\varphi \mathrel{-\!\!*} \psi$ (4.2) *is monotonic in any DCSA*

Here, these two definitions are justified by theorem 1. Moreover, a separation algebra with a discrete order is UCSA and DCSA. Also UCSA and DCSA are closed under cartesian product, disjoint sum and exponentiation.

Unfortunately, many separation algebras don't satisfy requirements 10 and 11. For instance, the following example describes a heap where short is two bytes long. It is motivated by the treatment of multibyte locks in VST [1, pp. 366-7] .

Example 1 (Typed Heaps). Let a typed heap be $\mathcal{H}_\mathcal{T} : \mathbb{N} \rightharpoonup \{\mathsf{char}, \mathsf{short}_1, \mathsf{short}_2\}$, such that $H(n) = \mathsf{short}_1$ iff $H(n+1) = \mathsf{short}_2$. Define $\oplus$ as the typical heap addition (i.e. disjoint map union) and define the order as follows

$$H_1 \sqsubseteq_{\mathcal{H}_\mathcal{T}} H_2 \triangleq \forall n \in \mathsf{dom}\ H_1,\ n \in \mathsf{dom}\ H_2 \text{ and } H_1(n) = H_2(n) \text{ or } H_1(n) = \mathsf{char}$$

Then $(\mathcal{H}_\mathcal{T}, \sqsubseteq_{\mathcal{H}_\mathcal{T}}, \oplus)$ is an OSA which is DCSA but not UCSA.

In order to be as inclusive as possible, it is desirable to avoid requirements 10 and 11. With that goal, stronger semantics of $*$ and $\mathrel{-\!\!*}$ have been proposed [17] [31] to be monotonic by design. Theorem 2 justifies the choice.

Definition 12 (Strong semantics).

$$\begin{aligned} m \vDash \varphi * \psi \triangleq\ & \exists\ m_0,\ m_1,\ m_2\ \textit{s.t.}\ \ m_0 \leq m, \oplus(m_1, m_2, m_0) \\ & \textit{and}\ \ m_1 \vDash \varphi\ \textit{and}\ m_2 \vDash \psi \end{aligned} \tag{12.1}$$

$$\begin{aligned} m \vDash \varphi \mathrel{-\!\!*} \psi \triangleq\ & \textit{for any}\ m_0,\ m_1,\ m_2\ \textit{if}\ m \leq m_0, \oplus(m_0, m_1, m_2) \\ & \textit{then}\ m_1 \vDash \varphi\ \textit{implies}\ m_2 \vDash \psi \end{aligned} \tag{12.2}$$

Theorem 2 (Strong semantics monotonicity). *For any assertion φ and ψ, if their denotations are both monotonic, then*

1. *strong semantics of separating conjunction* (12.1) *ensures the denotation of $\varphi * \psi$ to be monotonic*
2. *strong semantics of separating implication* (12.2) *ensures the denotation of $\varphi -\!\!* \psi$ to be monotonic.*

Alas, it is not always possible to use both strong semantics (12.1) and (12.2). These definitions enforce monotonicity, but might break other proof rules. As we show with the following example, separation logic becomes unsound w.r.t such strong semantics in the OSAs that are closed in neither direction.

Example 2. Consider the following separation algebra:

- Let M be $\{\bot, a_1, a_2, b_1, b_2, \top\}$.
- Let $\oplus$ be $\{(a_1, a_1, a_2), (b_1, b_1, b_2)\}$.
- Let $\leq$ be $\{(\bot, a_1), (a_2, b_1), (b_2, \top)\} \cup \{(s, s) | s \in M\}$

On this model, $*$ASSOC is unsound w.r.t. the semantics of $*$ and $-\!\!*$ defined by (12.1) and (12.2).

Consequently, one must accept at least one restriction (i.e. upwards-closed or downwards-closed) over OSAs. There are only three viable semantics, each paired with a family of OSAs:

Definition 13 (Separation Logic semantics). *Over the propositional connectives (i.e. $\wedge$, $\vee$, $\to$, $\bot$), the semantics of separation logic is defined by definition 6. The semantics of the separating connectives is then defined as follows*

1. *In* upwards semantics *($\cdot \models^{\Uparrow} \cdot$), which is defined on upwards closed OSAs, the conjunction is weak and implication strong (i.e. 4.1 and 12.2);*
2. *In* downwards semantics *($\cdot \models^{\Downarrow} \cdot$), which is defined on downwards closed OSAs, the conjunction is strong and implication weak (i.e. 12.1 and 4.2).*
3. *In* flat semantics *($\cdot \models^{=} \cdot$), which is defined on OSAs closed in two directions, both separating connectives are weak (i.e. 4.1 and 4.2).*

Lemma 1. *If a separation algebra is upwards-closed, the weak semantics of $*$ on it is equivalent to the strong semantics. If a separation algebra is downwards-closed, the weak semantics of $-\!\!*$ on it is equivalent to the strong semantics.*

Corollary 1. *For separation algebras with discrete order, upwards semantics, downwards semantics and flat semantics are equivalent.*

A *flat semantics* is by far the most common and intuitive; it is used whenever the algebra is discrete, as suggested by corollary 1, and was used in Reynolds's original logic [32]. But in many applications, there is no natural flat semantics since the underlying model is not closed in both ways. The *upwards* semantics is used by Dockins *et al.* [13]. The *downwards* semantics appears in [15] [31] [24].

To the best of our knowledge, all semantics are covered by definition 13, although in some cases it is not immediately obvious [4] [5] [6]. The followings are typical examples.

Example 3 (Step-indexed heap). Consider a heap (unordered) separation algebra $(\mathcal{H}, \oplus_{\mathcal{H}})$, the following is a semantics of separation logic defined on $\mathbb{N} \times \mathcal{H}$:

$$(i,h) \vDash \varphi * \psi \triangleq \exists\ h_1\ h_2 \text{ s.t. } \oplus_{\mathcal{H}}(h_1,h_2,h) \text{ and } (i,h_1) \vDash \varphi \text{ and } (i,h_2) \vDash \psi$$

$$(i,h) \vDash \varphi \mathrel{-\!\!*} \psi \triangleq \text{ for any } j,\ h_1 \text{ and } h_2 \text{ if } i \geq_{\mathbb{N}} j \text{ and } \oplus(h,h_1,h_2) \text{ then } (j,h_1) \vDash \varphi \text{ implies } (j,h_2) \vDash \psi$$

Such models are used to define mixvariant recursive predicates [3] inside heaps.

In fact, this semantics on step indexed heap is just an upwards semantics on a product algebra, as formalized in proposition 1. It seems like a hybrid semantics: upwards for the indices (in the inverted order of naturals) and flat for the heap. But since the monotonic heap is upwards-closed, by lemma 1, such hybrid semantics is equivalent to an upwards semantics.

Proposition 1. *The semantics defined in ex.3 is equivalent to upwards semantics on the cartesian product of the index algebra* $(i.e. \mathbb{N}, \geq_{\mathbb{N}}, \Delta_3)$ *and the heap* $(i.e. \mathcal{H}, \sqsubseteq_{\mathcal{H}}, \oplus_{\mathcal{H}})$*. Here,* $\Delta_3 \triangleq \{(n,n,n) | n \in \mathbb{N}\}$.

The following example is first used in FSCQ [11] to reason about a file system in which the hardware may crush. This separation logic is so powerful that its developer has verified a crush-recovery program even if a crush may happen again in the recovery process. In their separation logic, an assertion is required to be invariant w.r.t. buffer flushing, i.e. if an assertion is true before flushing then it should be true after flushing.

Example 4 (Heap with write buffer). Let V^+ be the set of non-empty lists of V. Such a nonempty list of values can represent one storage location with a write buffer. We call $l' \in V^+$ a flushing result of $l \in V^+$ if l' is a suffix of l, i.e. exists l'' such that $l = l''l'$. A heap with write buffer is an OSA $(\mathcal{B}, \oplus_{\mathcal{B}}, \leq_{\mathcal{B}})$:
$\mathcal{B} \triangleq \mathsf{adr} \rightharpoonup V^+$, $\oplus_{\mathcal{B}}$ is heap-join relation and $\leq_{\mathcal{B}} \triangleq \{(b_1,b_2) \mid \mathsf{dom}\ b_1 = \mathsf{dom}\ b_2 \text{ and } \forall x \in \mathsf{dom}\ b_1.\ \exists l.\ b_1(x) = l\ b_2(x)\}$

This OSA is upwards closed and downwards closed, and the separational logic defined by FSCQ is actually the flat semantics on it. The following variants of it show the flexibility of our framework.

Example 5 (More refined buffer flushing). (1) $\preccurlyeq_{\mathcal{B}} \triangleq \{(b_1,b_2) \in \leq_{\mathcal{B}} \mid \forall x \in \mathsf{dom}\ b_1.$ either $b_1(x) = b_2(x)$ or $b_2(x)$ is a singleton list$\}$. This denies any partial flushing on every single location. (2) $\sqsubseteq_{\mathcal{B}} \triangleq \{(b_1,b_2) \in \leq_{\mathcal{B}} \mid$ either $\forall x \in \mathsf{dom}\ b_1.\ b_1(x) = b_2(x)$ or $\forall x \in \mathsf{dom}\ b_1.\ b_2(x)$ is a singleton list$\}$. This denies any partial flushing on the whole heap.

The OSA defined by the first variant is still closed in both directions. However, the second variant is only upwards closed. Thus, only upwards semantics can be defined on it.

5.4 Semantic equivalence

For now, we have defined upwards semantics on upwards closed OSAs, downwards semantics on downwards closed OSAs, and have defined flat semantics on OSAs closed in both directions.

In this subsection, we show that they are all equivalent to instances of each other. This is not to say that we can define flat semantics on OSAs with only one closed property. Instead, we demonstrate a practical way of converting an upwards closed *or* downwards closed OSA into an upwards closed *and* downwards closed OSA. We show an equivalence between the flat semantics on the new OSA and the upwards or downwards semantics on the original OSA.

As our first step, we define the following two model transformations:

Definition 14 (Upwards closure and downwards closure). *Given a separation algebra $(M, \leq, \oplus)$, its upwards closure is the triple $(M, \leq, \oplus^{\Uparrow})$ where $\oplus^{\Uparrow}(m_1, m_2, m)$ iff there is m' such that $m' \leq m$ and $\oplus(m_1, m_2, m')$.*

Given a separation algebra $(M, \leq, \oplus)$, its downwards closure is the triple $(M, \leq, \oplus^{\Downarrow})$ where $\oplus^{\Downarrow}(m_1, m_2, m)$ iff there are m_1' and m_2' such that $m_1 \leq m_1'$, $m_2 \leq m_2'$ and $\oplus(m_1', m_2', m)$.

For example, the downwards closure of index algebra (*i.e.*$\mathbb{N}, \geq_{\mathbb{N}}, \Delta_3$) is the *minimum algebra* $(\mathbb{N}, \leq_{\mathbb{N}}, \oplus_{\min})$, with $\oplus_{\min} \triangleq \{(x, y, z) \mid z \leq_{\mathbb{N}} x \text{ and } z \leq_{\mathbb{N}} y\}$.

Lemma 2. *Given an ordered separation algebra $(M, \leq, \oplus)$: If $(M, \leq, \oplus)$ is upwards closed, $(M, \leq, \oplus^{\Downarrow})$ is a downwards closed and upwards closed ordered separation algebra. If $(M, \leq, \oplus)$ is downwards closed, $(M, \leq, \oplus^{\Uparrow})$ is a downwards closed and upwards closed ordered separation algebra.*

At this point, you might think that any OSA can be made downwards- and upwards-closed by taking both of its closures. Unfortunately, such a two-sided closure might not be an ordered separation algebra at all!

Proposition 2. *Let $(M, \leq, \oplus^{\Downarrow\Uparrow})$ be the upwards closure of the downwards closure of example 2. Then $(M, \leq, \oplus^{\Downarrow\Uparrow})$ is not associative.*

When the algebra is upwards- or downwards-closed the closure is upwards- and downwards-closed. Then, we can use the flat semantics on closures, which happens to be equivalent with a stronger semantics over the original algebra:

Theorem 3. *Given an extended Kripke model $\mathcal{M} = (M, \leq, \oplus, J)$*

1. *if it is downwards closed, then the flat semantics on $\mathcal{M}^{\Uparrow}$ is equivalent to the downwards semantics on $\mathcal{M}$, i.e. for any φ and m, $m \vDash^{=}_{\mathcal{M}^{\Uparrow}} \varphi$ iff $m \vDash^{\Downarrow}_{\mathcal{M}} \varphi$*
2. *if it is upwards closed, then the flat semantics on $\mathcal{M}^{\Downarrow}$ is equivalent to the upwards semantics on $\mathcal{M}$, i.e. for any φ and m, $m \vDash^{=}_{\mathcal{M}^{\Downarrow}} \varphi$ iff $m \vDash^{\Uparrow}_{\mathcal{M}} \varphi$*

Here, $\mathcal{M}^{\Uparrow} \triangleq (M, \leq, \oplus^{\Uparrow}, J)$ and $\mathcal{M}^{\Downarrow} \triangleq (M, \leq, \oplus^{\Downarrow}, J)$.

Recall that flat semantics are direct instances of upwards semantics and downwards semantics: the theorem follows as a corollary of lemma 1.

Theorem 4. *Given an extended Kripke model* $\mathcal{M} = (M, \leq, \oplus, J)$ *with downwards closed and upwards closed separation algebra, for any* φ *and* m*: (1)* $m \vDash^{=}_{\mathcal{M}} \varphi$ *iff* $m \vDash^{\Uparrow}_{\mathcal{M}} \varphi$ *(2)* $m \vDash^{=}_{\mathcal{M}} \varphi$ *iff* $m \vDash^{\Downarrow}_{\mathcal{M}} \varphi$

Thus, in summary, flat semantics is a direct instance of upwards semantics and downwards semantics (thm. 4). Upwards semantics and downwards semantics are instances of flat semantics via the closures (thm. 3).

As far as we know, the idea of our model transformation and the semantic preservation is completely novel. However, we consider Atkey's separation logic for amortized resource analysis [4] a precursor worth mentioning. To use separation logic for amortized resource analysis, Atkey used the cartesian product of a discrete heap and a *consumable* resource (which is just an ordered monoid). The author wanted to use the usual and simpler flat semantics, but this is impossible because ordered monoids are not necessarily upwards-closed. To solve the problem, he defined an unusual separation algebra that is equivalent to the product of a discrete heap and the closure of the resource.

Example 6 (Resource-bounds). Suppose $(\mathcal{R}, \oplus_{\mathcal{R}}, \leq_{\mathcal{R}})$ is an ordered commutative monoid (for any $r_1 \leq_{\mathcal{R}} r_1'$ and $r_2 \leq_{\mathcal{R}} r_2'$, $r_1 \oplus_{\mathcal{R}} r_2 \leq_{\mathcal{R}} r_1' \oplus_{\mathcal{R}} r_2'$), and $(\mathcal{H}, \oplus_{\mathcal{H}})$ is an unordered heap. The following OSA on the product type, $(\mathcal{R} \times \mathcal{H}, \oplus_{\mathcal{R}\times\mathcal{H}}, \leq_{\mathcal{R}\times\mathcal{H}})$, is upwards closed and downwards closed.

$$\leq_{\mathcal{R}\times\mathcal{H}} \triangleq \{(i,h),(j,h) \mid i \leq_{\mathcal{R}} j\}$$
$$\oplus_{\mathcal{R}\times\mathcal{H}} \triangleq \{(i_1,h_1),(i_2,h_2),(i_3,h_3) \mid \oplus_{\mathcal{H}} (h_1,h_2,h_3) \text{ and } i_1 \oplus_{\mathcal{R}} i_2 \leq_{\mathcal{R}} i_3\}$$

This OSA is equivalent with $(\mathcal{R}, \oplus_{\mathcal{R}}, \leq_{\mathcal{R}})^{\Uparrow} \times (\mathcal{H}, =, \oplus_{\mathcal{H}})$.

5.5 Semantics of emp

Garbage-collect separation logic and malloc/free separation logic disagree on the semantics of emp. In a malloc/free separation logic,

$$m \vDash \mathsf{emp} \quad \triangleq \quad m \text{ is a unit} \tag{Emp$_1$}$$

and for a garbage-collected separation logic, emp just means True, i.e.

$$m \vDash \mathsf{emp} \quad \triangleq \quad \text{always} \tag{Emp$_2$}$$

We propose the following unification of the semantics of emp:

Definition 15 (Semantics of emp). *For a separation algebra* $(M, \leq, \oplus)$ *and* $m \in M$*,* $m \vDash \mathsf{emp}$ *iff* m *is increasing*

Definition 15 will not be sound if the set of increasing elements is not monotonic. To solve that, just like we did in section 5.3, we can define a stronger semantics to ensure soundness. This stronger semantics is an instance of definition 15 via the downwards closure, just like in theorem 3. Nevertheless, we know of no practical application of such semantics, we omit the discussion here. In what follows we just assume that the increasing set is monotonic.

Lemma 3. *If the algebra is downwards-closed, the increasing set is monotonic.*

For increasing algebras, all elements are increasing, so the semantics of emp is equivalent to (Emp_2). For algebras with discrete order, as stated in the lemma below, only units are increasing, so the semantics of emp is equivalent to (Emp_1)

Lemma 4. *In a separation algebra with discrete order, an element is increasing iff it is a unit.*

Finally, we define a *unital separation algebra* as one in which each element has a "increasing part".

Definition 16 (Residue). *In a separation algebra $(M, \leq, \oplus)$, we say m is a* residue *of n if there exists n' such that $\oplus(m, n', n)$ and $n \leq n'$.*

Definition 17 (Unital separation algebras). *A separation algebra is* unital *if all elements have an increasing residue. For antisymmetric orders this is equivalent to: all elements have a identity element.*

Unital OSAs are closed under product, sum and exponentiation.

6 Parametric Soundness And Completeness

It is known that different propositional logics are sound and complete with respect to their corresponding class of Kripke models. Intuitionistic logic is sound and complete w.r.t. Kripke semantics in all models [23]; Classical logic is sound and complete w.r.t. Kripke semantics in the Kripke models with a discrete order.

In this section, we determine corresponding classes of extended Kripke models for each separation logic, as defined in section 3, based on the framework for separation algebras developed in section 5. We then proceed to prove soundness and completeness for each separation logic w.r.t the flat semantics in it's corresponding class of models. According to the equivalence theorem proved in section 5.4, separation logics are then also proven sound and complete w.r.t. upwards semantics and downwards semantics.

Definition 18. *An order $\leq$ is* nonbranching *iff for any m, n and n', with $m \leq n$ and $m \leq n'$ then $n \leq n'$ or $n' \leq n$.*

An order $\leq$ always-joins *iff for any m, n and n', if $m \leq n$ and $m \leq n'$ then exists n'' s.t. $n \leq n''$ and $n' \leq n''$.*

Definition 19. *An OSA has its* increasing elements self-joining *iff for any m, if it is increasing then $\oplus(m, m, m)$.*

An OSA has normal increasing elements *iff for any n_1, n_2 and m, if $n_1 \oplus n_2 = mm$ is increasing then $n_1 \leq m$. In other words, an increasing element can only be split into smaller pieces.*

Definition 20 (Corresponding class of extended Kripke models). *Given a separation logic Γ, its* corresponding class of extended Kripke models *are the set of models which (1) are upwards-closed (2) are downwards-closed (3) are unital (4) satisfies the canonical properties of all optional axioms in Γ, as listed in this table:*

Optional axiom	*Canonical property*
EM $\in \Gamma$	*Discrete order*
GD $\in \Gamma$	*Nonbranching order*
WEM $\in \Gamma$	*Always-joining order*
$*$E $\in \Gamma$	*Increasing*
eDUP $\in \Gamma$	*Increasing elements self-joining*
eE $\in \Gamma$	*Have normal increasing elements*

Theorem 5 (Parametric soundness and completeness). *A separation logic Γ is sound and complete w.r.t. the flat semantics in Γ's corresponding class of models.*

Here, soundness and completeness are defined in the usual way. The soundness proof is trivial. We establish completeness by a Henkin-style proof. An interesting thing here is that the canonical model (constructed by derivable-closed disjunction-witnessed consistent sets of assertions) is upwards closed and downwards closed. This in some sense justifies our unifying solution. We put a detailed proof in an online appendix.

Remark. The correspondences between GD and nonbranching order, and between WEM and always-joining order, are standard results in propositional logics. But we find separation logic instances that admitted these intermediate rules unbeknownst to the authors For example, the step-indexed model of VST [1] has a nonbranching order so its separation logic should have GD. The order used in FSCQ [11] is always-joining so its separation logic should have WEM.

7 Applications Of The Unifying Semantics

In the past 15 years, separation logic has been prolific tool for modular program verification. However, until now, most research was only applicable to one specific semantic model. For example, Ishtiaq and O'Hearn [19] showed that the frame rule is sound as long as the operational semantics has the frame property, but their conclusion was only demonstrated for unordered separation algebras (in our framework, separation algebra with discrete order). So, their work does not directly benefit separation-logic-based verification tools such as VST [1] and Iris [22], since their semantics are based on step-indexed models. Similarly, the soundness of CSL [7] was only established on unordered separation algebra.

We have already shown one example of how to generalize results about separation logic. The soundness proof of IP + SL, in section 6, holds for all semantics of separation logic (because they are instances of the flat semantics). In this sections, we will further shows that the two soundness results mentioned above— frame rule and CSL— can be generalized to any ordered separation algebra with flat semantics, and thus for all separation logics.

7.1 Frame rule

The frame is a fundamental rule for modular reasoning in separation logic.

$$\frac{\{\varphi\}\ c\ \{\psi\}}{\{\varphi * \xi\}\ c\ \{\psi * \xi\}}\ (\text{FRAME})$$

Ishtiaq and O'Hearn [19] showed that the frame rule is sound with their "classical" separation logic, as long as the operational semantics has, what the called, *the frame property*. Here, we generalize the definition of frame property for all OSAs, and show that the soundness of frame rule still holds.

Definition 21 (Small-step semantics). *A small-step semantics of a programming language is a tuple* $(M, \mathbf{cmd}, \rightsquigarrow)$ *in which* M *represents program states,* **cmd** *represents program commands and* $\rightsquigarrow$ *is a binary relation between* $(M \times \mathbf{cmd})$ *and* $(M \times \mathbf{cmd} + \{\mathbf{Err}, \mathbf{NT}\})$*. Here,* **Err** *and* **NT** *means error state and nonterminating state respectively.*

We write $\rightsquigarrow^*$ *to mean the transitive reflexive closure of* $\rightsquigarrow$

Definition 22 (Accessibility from small-step semantics). *Given a small-step semantics* $(M, \mathbf{cmd}, \rightsquigarrow)$*, we define accessibility as a binary relation between* M *and* $M + \{\mathbf{Err}, \mathbf{NT}\}$*:*

1. $m \xrightarrow{c} m'$ *iff* $(m, c) \rightsquigarrow^* (m', \text{skip})$
2. $m \xrightarrow{c} \mathbf{Err}$ *iff* $(m, c) \rightsquigarrow^* \mathbf{Err}$
3. $m \xrightarrow{c} \mathbf{NT}$ *iff* $(m, c) \rightsquigarrow^* \mathbf{NT}$ *or there exist infinite sequences* $\{m_k\}$ *and* $\{c_k\}$ *(for* $k \in \mathbb{N}$*) such that* $(m, c) \rightsquigarrow (m_0, c_0)$ *and* $(m_k, c_k) \rightsquigarrow (m_{k+1}, c_{k+1})$*.*

Definition 23 (Frame property). *An accessibility relation* $(\cdot \xrightarrow{c} \cdot)$ *satisfies frame property w.r.t. an ordered separation algebra* $(M, \leq, \oplus)$ *if for all* $m, m_f, n, n' \in M$ *and any command* c*,*

1. *if* $\oplus(m, m_f, n)$ *and* $n \xrightarrow{c} \mathbf{Err}$*, then* $m \xrightarrow{c} \mathbf{Err}$
2. *if* $\oplus(m, m_f, n)$ *and* $n \xrightarrow{c} \mathbf{NT}$*, then* $m \xrightarrow{c} \mathbf{NT}$
3. *if* $\oplus(m, m_f, n)$*,* $n \xrightarrow{c} n'$ *and executing* c *terminates normally from* m*, then there exists* $m'_f, m' \in M$ *s.t.* $m \xrightarrow{c} m'$*,* $\oplus(m', m'_f, n')$ *and* $m_f \leq m'_f$

Definition 24 (Validity of Hoare triple). *Given a small-step semantics, a Hoare triple* $\{\varphi\}\ c\ \{\psi\}$ *is valid if and only if for any* $m \in M$*, if* $m \vDash^= \varphi$ *then (1)* $m \not\xrightarrow{c} \mathbf{Err}$ *(2) for any* n*, if* $m \xrightarrow{c} n$ *then* $n \vDash^= \psi$*.*

Theorem 6 (Soundness of frame rule). *If the operational semantics satisfies the frame property w.r.t. an ordered separation algebra* $(M, \leq, \oplus)$ *which is upwards closed and downwards closed, then the frame rule is sound, i.e. if* $\{\varphi\}\ c\ \{\psi\}$ *is valid then* $\{\varphi * \chi\}\ c\ \{\psi * \chi\}$ *is valid.*

7.2 Concurrent separation logic

Concurrent separation logic is an extension of separation Hoare logic to reason about concurrent programs. Brookes [7] proved CSL to be sound for unordered heap models; in particular, he required cancellativity of the algebra. But separation logics with ghost resources [22] don't have cancellative algebras: the ghosts don't naturally cancel. Is CSL with ghost resources sound? As we show, the Brookes's soundness proof of CSL can be generalized to any models of ordered separation algebra, even without cancellativity.

Brookes defines the behavior of his imperative programming language (with concurrent primitives) via trace semantics.

Definition 25 (Trace semantics). *A trace semantics of a programming language is defined by a program state set M, a set R of resource identifiers, a command denotation function $[\![\cdot]\!]$ and a enable relation $\cdot \rightsquigarrow_a \cdot$. Specifically, for any program c, its denotation $[\![c]\!]$ is a set of traces; every trace is a finite or infinite list of actions. For any action a, $\cdot \rightsquigarrow_a \cdot$ is a relation between $\mathcal{P}(R) \times M$ and $\mathcal{P}(R) \times M + \{\mathbf{Err}\}$.*

In all actions, two kinds are special: $\mathrm{rel}(r)$ and $\mathrm{acq}(r)$ which means releasing and acquiring a resource r respectively. All other actions are nonresource actions. Actions' behavior satisfies the following properties:

$(A, m) \rightsquigarrow_{\mathrm{rel}(r)} (B, n)$ iff $r \notin B, A = B \cup \{r\}$ and $m = n$

$(A, m) \rightsquigarrow_{\mathrm{acq}(r)} (B, n)$ iff $r \notin A, B = A \cup \{r\}$ and $m = n$

$(A, m) \not\rightsquigarrow_{\mathrm{rel}(r)} \mathbf{Err}$ and $(A, m) \not\rightsquigarrow_{\mathrm{acq}(r)} \mathbf{Err}$

If a is a nonresource action and $(A, m) \rightsquigarrow_a (B, n)$ then $A = B$

If a is a nonresource action,$\cdot \rightsquigarrow_a \cdot$ satisfies a frame property like definition 23.

From a trace semantics, we can define the validity of guarded Hoare triples following Brookes's method.

Definition 26 (Thread-local enable relation). *Suppose Γ is a partial function from resource identifiers to their resource invariants and the program state is the underlying set of an upwards closed and downwards closde OSA, $\mathcal{M} = (M, \leq, \oplus)$, then the thread-local enable relation $\cdot \rightsquigarrow_{a,\Gamma} \cdot$ is defined as follows:*

1. $(A, m) \rightsquigarrow_{\mathrm{rel}(r),\Gamma} (B, n)$ *iff* $r \notin B, A = B \cup \{r\}$ *and exists* f *such that* $f \models^{=} \Gamma(r)$ *and* $\oplus(m, f, n)$
2. $(A, m) \not\rightsquigarrow_{\mathrm{rel}(r),\Gamma} \mathbf{Err}$
3. $(A, m) \rightsquigarrow_{\mathrm{acq}(r),\Gamma} (B, n)$ *iff* $r \notin A, B = A \cup \{r\}$ *and* n *is the greatest element in* $\{n \mid$ *exists* f *such that* $f \models^{=} \Gamma(r)$ *and* $\oplus(n, f, m)\}$
4. $(A, m) \rightsquigarrow_{\mathrm{acq}(r),\Gamma} \mathbf{Err}$ *iff* $r \notin A$ *and there is no* n *and* f *such that* $f \models^{=} \Gamma(r)$ *and* $\oplus(n, f, m)$
5. *If* a *is a nonresource action,* $\cdot \rightsquigarrow_{a,\Gamma} \cdot$ *is the same as* $\cdot \rightsquigarrow_a \cdot$

Definition 27 (Accessibility from trace semantics). *For any invariant mapping Γ, any trace t, $\alpha \in \mathcal{P}(R) \times M$ and $\beta \in \mathcal{P}(R) \times M + \{\mathbf{Err}, \mathbf{NT}\}$, $\alpha \xrightarrow[\Gamma]{c} \beta$ iff exists $t \in [\![c]\!]$ such that $\alpha \rightsquigarrow^*_{t,\Gamma} \beta$. Here $\cdot \rightsquigarrow^*_{t,\Gamma} \cdot$ is defined by $\cdot \rightsquigarrow_{a,\Gamma} \cdot$ like definition 22.*

We call $\cdot \xrightarrow[\Gamma]{c} \cdot$ the thread-local accessibility relation. Similarily, we define $\cdot \xrightarrow{c} \cdot$ from the (nonlocal) enable relation $\cdot \rightsquigarrow_a \cdot$. We omit the details here.

Definition 28 (Validity of guarded Hoare triple). *A guarded Hoare triple $\Gamma \vdash \{\varphi\}\ c\ \{\psi\}$ is valid if and only if for any $m \in M$, if $m \vDash^= \varphi$ then (1) $(\{\}, m) \not\xrightarrow[\Gamma]{c} \mathbf{Err}$ (2) for any n, if $(\{\}, m) \xrightarrow[\Gamma]{c} (\{\}, n)$ then $n \vDash^= \psi$.*

Brookes showed that if $\Gamma \vdash \{\varphi\}\ c\ \{\psi\}$ is valid and $m \vDash^= \varphi$ then (1) $(\{\}, m) \not\xrightarrow{c} \mathbf{Err}$ (2) for any n, if $(\{\}, m) \xrightarrow{c} (\{\}, n)$ then $n \vDash^= \psi$. This means: although triples' validity is defined on a thread-local view, it ensures global safety.

Another preparation before presenting Hoare rules is defining the concept of *preciseness*. In heap model, an assertion φ is precise when inside any memory m there is at most one submemory m_1 which satisfies φ. The following is our generalization. It is reasonable since it is the original preciseness on any cancellative separation algebra with discrete order.

Definition 29 (Preciseness). *Given an upwards closed and downwards closed separation algebra $(M, \leq, \oplus)$, an assertion φ is precise iff for any m, if $S \triangleq \{m_1 \mid \text{ exists } m_2 \text{ such that } \oplus(m_1, m_2, m), m_2 \vDash^= \varphi\}$ is nonempty then S has a greatest element.*

We are now ready to state our generalized theorem for the soundness of CSL.

Theorem 7 (Soundness of CSL). *If all resource invariants are precise, then Hoare rules in CSL are sound w.r.t. flat semantics of separation logic. The following are the Hoare rules of concurrent primitives:*

1. *If $\Gamma \vdash \{\varphi_1\}c_1\{\psi_1\}$ and $\Gamma \vdash \{\varphi_2\}c_2\{\psi_2\}$, then $\Gamma \vdash \{\varphi_1 * \varphi_2\}\ c_1||c_2\ \{\psi_1 * \psi_2\}$*
2. *If $\Gamma \vdash \{\varphi * \xi\}\ c\ \{\psi * \xi\}$ and r does not freely occur in c, then $\Gamma; r : \xi \vdash \{\varphi\}$ with r do c $\{\psi\}$*

Here, $[\![c_1||c_2]\!]$ is the resource coherent interleaving of $[\![c_1]\!]$ and $[\![c_2]\!]$, and $[\![\text{with } r \text{ do } c]\!]$ is acquiring r, applying $[\![c]\!]$ and releasing r.

The proof follows the same lines as Brookes's proof. We formalize it in Coq and omit it here.

8 Future work

We are particularly excited to use the present work as a starting point to find a unifying framework of Hoare separation logic. There are many incompatible

definitions for the semantics for Hoare triples [22,1,19] which make their different results incompatible. We believe the present work is the first step towards resolving this incompatibility, but there are many issues yet to be solved. For instance, the two examples in section 7 have first order Hoare logics, but Hoare logics for programs with function calls are high ordered. Unifying such Hoare logics is more challenging. It will also be particularly challenging to unify the operational semantics for verifying concurrent programs. We believe this is the way to solve Parkinson's challenge for the next 700 separation logics.

Our hope is that the present work will be a fertile ground for generalizing many known results, as we did in section 7. Of particular interest to us is whether *precisesness* is required in a Hoare logic with the conjunction rule [26], [16]. A related conundrum is whether a separation algebra must be cancellative in order to have the conjunction rule. Both questions have been answered for classical, malloc/free logics, but are open in general.

9 Conclusion

We have clarified the terminology "classical vs. intuitionistic" and "malloc/free vs. garbage-collected" separation logic. They are two independent taxonomies.

We present flat semantics on upwards-closed and downwards-closed OSA as a unification for all different semantics of separation logics. All separation logics are proved sound and complete w.r.t. corresponding model classes. This unification is powerful enough to generalize related concepts like *frame property* and *preciseness* and to generalize theoretical applications of separation logic like the soundness of frame rule (given frame property) and the soundness of CSL.

All the definitions, propositions, lemmas and theorems in this paper have been formalized in Coq. We also put a detailed paper proof of two main theorems of this paper in an appendix: semantic equivalence theorem (thm. 3) and completeness theorem (thm. 5). Our Coq development and paper proofs are both accessible online[2].

Acknowledgment. This research was supported in part by NSF Grant CCF-1521602.

References

1. Andrew W. Appel, Robert Dockins, Aquinas Hobor, Lennart Beringer, Josiah Dodds, Gordon Stewart, Sandrine Blazy, and Xavier Leroy. *Program Logics for Certified Compilers*. Cambridge, 2014.
2. Andrew W. Appel and David A. McAllester. An indexed model of recursive types for foundational proof-carrying code. *ACM Trans. Program. Lang. Syst.*, 23(5):657–683, 2001.

[2] Coq development: `https://github.com/QinxiangCao/UnifySL`. Appendix: `http://www.cs.princeton.edu/~appel/papers/bringing-order-appendix.pdf`.

3. Andrew W. Appel, Paul-André Melliès, Christopher D. Richards, and Jérôme Vouillon. A very modal model of a modern, major, general type system. In *Proceedings of the 34th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, 2007.
4. Robert Atkey. Amortised resource analysis with separation logic. *Logical Methods in Computer Science*, 7(2), 2011.
5. Jesper Bengtson, Jonas Braband Jensen, Filip Sieczkowski, and Lars Birkedal. Verifying object-oriented programs with higher-order separation logic in coq. In *Interactive Theorem Proving - Second International Conference*, 2011.
6. Lars Birkedal, Bernhard Reus, Jan Schwinghammer, Kristian Støvring, Jacob Thamsborg, and Hongseok Yang. Step-indexed Kripke models over recursive worlds. In *Proceedings of the 38th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, 2011.
7. Stephen D. Brookes. A semantics for concurrent separation logic. In *CONCUR 2004 - Concurrency Theory, 15th International Conference, London, UK, August 31 - September 3, 2004, Proceedings*, pages 16–34, 2004.
8. James Brotherston and Max Kanovich. Undecidability of propositional separation logic and its neighbours. In *Logic in Computer Science (LICS), 2010 25th Annual IEEE Symposium on*, pages 130–139. IEEE, 2010.
9. James Brotherston and Jules Villard. Parametric completeness for separation theories. In *The 41st Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, 2014.
10. Cristiano Calcagno, Peter W. O'Hearn, and Hongseok Yang. Local action and abstract separation logic. In *Proceedings of the 22Nd Annual IEEE Symposium on Logic in Computer Science*, LICS '07, pages 366–378, Washington, DC, USA, 2007. IEEE Computer Society.
11. Haogang Chen, Daniel Ziegler, Tej Chajed, Adam Chlipala, M. Frans Kaashoek, and Nickolai Zeldovich. Using crash hoare logic for certifying the FSCQ file system. In Ethan L. Miller and Steven Hand, editors, *Proceedings of the 25th Symposium on Operating Systems Principles, SOSP 2015, Monterey, CA, USA, October 4-7, 2015*, pages 18–37. ACM, 2015.
12. Thomas Dinsdale-Young, Lars Birkedal, Philippa Gardner, Matthew J. Parkinson, and Hongseok Yang. Views: compositional reasoning for concurrent programs. In *The 40th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, 2013.
13. Robert Dockins, Aquinas Hobor, and Andrew W. Appel. A fresh look at separation algebras and share accounting. In *Proceedings of the 7th Asian Symposium on Programming Languages and Systems*, APLAS '09, pages 161–177, Berlin, Heidelberg, 2009. Springer-Verlag.
14. Didier Galmiche and Dominique Larchey-Wendling. Expressivity properties of boolean BI through relational models. In *International Conference on Foundations of Software Technology and Theoretical Computer Science*, pages 357–368. Springer, 2006.
15. Didier Galmiche, Daniel Méry, and David J. Pym. The semantics of BI and resource tableaux. *Mathematical Structures in Computer Science*, 15(6):1033–1088, 2005.
16. Alexey Gotsman, Josh Berdine, and Byron Cook. Precision and the conjunction rule in concurrent separation logic. *Electr. Notes Theor. Comput. Sci.*, 276:171–190, 2011.
17. Aquinas Hobor, Robert Dockins, and Andrew W. Appel. A theory of indirection via approximation. In *Proceedings of the 37th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, 2010.

18. Chung-Kil Hur, Derek Dreyer, and Viktor Vafeiadis. Separation logic in the presence of garbage collection. In *Proceedings of the 26th Annual IEEE Symposium on Logic in Computer Science, LICS 2011, June 21-24, 2011, Toronto, Ontario, Canada*, pages 247–256, 2011.
19. Samin S. Ishtiaq and Peter W. O'Hearn. BI as an assertion language for mutable data structures. In *Conference Record of POPL 2001: The 28th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, 2001.
20. Jonas Braband Jensen. *Techniques for model construction in separation logic*. PhD thesis, IT University of Copenhagen, March 2014.
21. Jonas Braband Jensen and Lars Birkedal. Fictional separation logic. In *Programming Languages and Systems - 21st European Symposium on Programming*, 2012.
22. Ralf Jung, David Swasey, Filip Sieczkowski, Kasper Svendsen, Aaron Turon, Lars Birkedal, and Derek Dreyer. Iris: Monoids and invariants as an orthogonal basis for concurrent reasoning. In *Proceedings of the 42nd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, 2015.
23. Saul A. Kripke. Semantical analysis of intuitionistic logic i. In *Studies in Logic and the Foundations of Mathematics 50*, pages 92–130, 1965.
24. Dominique Larchey-Wendling and Didier Galmiche. Exploring the relation between intuitionistic BI and boolean BI: an unexpected embedding. *Mathematical Structures in Computer Science*, 19(3):435–500, 2009.
25. Peter W O'Hearn and David J Pym. The logic of bunched implications. *Bulletin of Symbolic Logic*, 5(2):215–244, 1999.
26. Peter W. O'Hearn, Hongseok Yang, and John C. Reynolds. Separation and information hiding. In *Proceedings of the 31st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2004, Venice, Italy, January 14-16, 2004*, pages 268–280, 2004.
27. Matthew J. Parkinson. The next 700 separation logics - (invited paper). In *Verified Software: Theories, Tools, Experiments, Third International Conference, VSTTE 2010, Edinburgh, UK, August 16-19, 2010. Proceedings*, pages 169–182, 2010.
28. Matthew J. Parkinson and Alexander J. Summers. The relationship between separation logic and implicit dynamic frames. In *Programming Languages and Systems - 20th European Symposium on Programming*, pages 439–458, 2011.
29. Alexandre Pilkiewicz and François Pottier. The essence of monotonic state. In *Proceedings of TLDI 2011: 2011 ACM SIGPLAN International Workshop on Types in Languages Design and Implementation*, pages 73–86, 2011.
30. François Pottier. Syntactic soundness proof of a type-and-capability system with hidden state. *J. Funct. Program.*, 23(1):38–144, 2013.
31. David J. Pym, Peter W. O'Hearn, and Hongseok Yang. Possible worlds and resources: the semantics of BI. *Theor. Comput. Sci.*, 315(1):257–305, 2004.
32. John C. Reynolds. Intuitionistic reasoning about shared mutable data structure. In *Millennial Perspectives in Computer Science*, pages 303–321. Palgrave, 2000.
33. Alex K Simpson. The proof theory and semantics of intuitionistic modal logic. Technical report, University of Edinburgh. College of Science and Engineering. School of Informatics., 1994.