A Semantic Foundation for Hidden State

Jan Schwinghammer[1], Hongseok Yang[2], Lars Birkedal[3], François Pottier[4], and Bernhard Reus[5]

[1] Saarland Univ. [2] Queen Mary Univ. of London [3] IT Univ. of Copenhagen
[4] INRIA [5] Univ. of Sussex

Abstract. We present the first complete soundness proof of the anti-frame rule, a recently proposed proof rule for capturing information hiding in the presence of higher-order store. Our proof involves solving a non-trivial recursive domain equation, and it helps identify some of the key ingredients for soundness.

1 Introduction

Information hiding, or *hidden state*, is one of the key design principles used by programmers in order to control the complexity of large-scale software systems. Being able to exploit this principle in the formal setting of a program logic could represent an important step towards the development of modular, scalable program verification techniques.

The idea is that an object (or function, or module) need not reveal in its interface the fact that it owns and maintains a private, mutable data structure. Hiding this internal invariant from the client has several beneficial effects. First, the complexity of the object's specification is slightly decreased. More importantly, the client is relieved from the need to thread the object's invariant through its own code. In particular, when an object has multiple clients, they are freed from the need to cooperate with one another in threading this invariant. Last, by hiding its internal state, the object escapes the restrictions on aliasing and ownership that are normally imposed to objects with mutable state.

It is worth emphasizing that *hiding* and *abstraction* (as studied, for instance, in separation logic [1,2,3]) are distinct mechanisms, which may co-exist within a single program logic.[1]

The recently proposed *anti-frame* proof rule [4] enables hiding in the presence of higher-order store (i.e., memory cells containing procedures or code fragments). In this paper, we study the semantic foundation of the anti-frame rule, and give the first complete soundness proof for it. Our proof involves the solution of an intricate recursive domain equation, and it helps identify some of the key ingredients for soundness.

[1] The abstraction is often implemented in terms of assertion variables (called abstract predicates by Parkinson) that describe private data structures of objects. These variables are exposed to a client, but their definitions are not, so that the object internals are presented to the client in an abstract form. The hiding, on the other hand, aims for concealing the object internals completely from the client.

Information hiding with frame and anti-frame rules Our results push the frontier of recent logic-based approaches to information hiding. These approaches adopt a standard semantics of the programming language, and deal with information hiding on a logical basis, by extending a Hoare calculus with special proof rules. These usually take the form of *frame rules* that allow the implementation of an object to ignore (hence implicitly preserve) some of the invariants provided by the context, and of *anti-frame rules*, which allow an object to hide its internal invariant from the context [5,6,7,4].

In its simplest form, the frame rule [5] states that invariants R can be added to valid triples: if $\{P\}C\{Q\}$ is valid, then so is $\{P * R\}C\{Q * R\}$, where the separating conjunction $P * R$ indicates that P and R govern disjoint regions of the heap. In subsequent developments, the rule was extended to handle higher-order procedures [6,7] and higher-order store [8,9]. Moreover, it was argued that both extensions of the rule support information hiding: they allow one to hide the invariant of a module [6] and to prove properties of clients, as long as the module is understood in continuation-passing style.

Thorough semantic analyses were required to determine the conditions under which these extensions of the frame rule are sound. Indeed, the soundness of these rules raises subtle issues. For instance, the frame rule for higher-order procedures turns out to be inconsistent with the conjunction rule, a standard rule of Hoare logic [6,7]. Furthermore, seemingly innocent variants of the frame rule for higher-order store have been shown unsound [9,10].

In the most recent development in this line of research, Pottier [4] proposed an anti-frame rule, which expresses the information hiding aspect of an object directly, instead of in continuation-passing style. Besides giving several extensive examples of how the anti-frame rule supports hidden state, Pottier argued that the anti-frame rule is sound by *sketching* a plausible syntactic argument. This argument, however, relied on several non-trivial assumptions about the existence of certain recursively defined types and recursively defined operations over types.

In this paper, we systematically study the semantic foundation of hidden state, as captured by frame and anti-frame rules, in the presence of higher-order store. In particular, we describe our soundness proof of a program logic that has both frame and anti-frame rules.

Overview of the technical development The anti-frame rule was originally proposed in an expressive type system for an ML-like language [4]. In the context of separation logic, it becomes an inference rule for deriving Hoare triples. A slightly simplified version of it takes the following form:

$$\frac{\{P \otimes R\}C\{(Q \otimes R) * R\}}{\{P\}C\{Q\}}$$

Recall that the separating conjunction $P' * Q'$ holds of a heap when the heap can be split into two sub-heaps that respectively satisfy P' and Q'. In order to specify properties of stored code, we allow assertions to contain nested triples [9], and introduce a $\otimes$ operator, whose meaning is roughly the following: $P' \otimes R'$ denotes

a version of P' where R' has been $*$-conjoined with the pre- and post-conditions of every triple, including deeply nested triples.

In the anti-frame rule above, the code C can be thought of as allocating and initializing an object. The assertion R describes an internal invariant of this object, which one wishes to hide. The conjunct $- * R$ in the post-condition of the premise ensures that the invariant R is established by C, so R holds initially. The two occurrences of $- \otimes R$ in the premise guarantee that every triple that appears in the premise has R as pre- and post-conditions, so every interaction between the object and the outside preserves R. The invariant R does not appear in the conclusion of the rule, so one reasons about the rest of the program just as if the object had no internal state.

Our soundness proof of the anti-frame rule is based on two key components. The first is a new interpretation of Hoare triples, which explicates the universal and existential quantifications that are implicit in the anti-frame rule. Let $P \circ R$ abbreviate $(P \otimes R) * R$. Roughly speaking, in our interpretation, a triple $\{P\}C\{Q\}$ is valid if, for all invariants R, the triple

$$\{P \circ R\}\, C\, \{\exists R'.\ Q \circ (R \circ R')\} \tag{1}$$

holds in the standard interpretation of triples. Pottier [4] showed how the anti-frame rule allows encoding ML-like weak references in terms of strong references. Readers who are familiar with models of ML references (see, e.g., [11]) may thus find the above interpretation natural. Roughly speaking, the code C has a function type $P \to Q$, whose interpretation is: "for all worlds R, if C is given an argument of type P in world R, then, for some future world $R \circ R'$ (an extension of R), C returns a result of type Q in world $R \circ R'$."

The second element in our soundness proof is a formalization of the above intuition. Our interpretation of assertions is parameterized by a set W of *worlds*, or *invariants*, so that, semantically, an *assertion* is a function $W \to \mathcal{P}(Heap)$ from worlds to (certain) sets of heaps. Corresponding to $R \circ R'$ in (1), there is a semantic operation $\circ$ on W that lets us combine two invariants. This operation induces a preorder on invariants, whereby $R \circ R'$ is greater than (i.e., a future world of) R.

In order to prove that the anti-frame rule is sound, we require assertions P to be monotonic with respect to the preorder on invariants: that is, $P(R)$ must be a subset of $P(R \circ R')$, for all P, R and R'. This lets us relate an assertion P at two different invariants R_0 and R_1, by first exhibiting an upper bound R_2 of R_0 and R_1 and then using the monotonicity to conclude $P(R_i) \subseteq P(R_2)$ for $i \in \{0, 1\}$. This forms an important step of our soundness proof.

In order to present our soundness proof as abstractly and elegantly as we can, we begin with an axiomatization of worlds and world composition, that is, we state a number of requirements that worlds should satisfy (Section 2). This allows us to define the interpretation of triples and establish some of its key properties in an abstract setting (Section 3).

In Sections 4 and 5, we move to a more concrete setting and present a small imperative programming language that features higher-order store, in the form

of storable commands. We equip it with a proof system, which features nested Hoare triples, frame rules, and anti-frame rules. As in Pottier's original setting, in this system, it is desirable that every assertion R be allowed to play the role of an invariant. As a consequence, in this concrete instance, the set of invariants W should be isomorphic to the semantic domain of assertions.

In summary, for this instance the requirements that we have described above amount to the following non-standard recursive domain equation:

$$\mathit{Assert} \;\cong\; \mathit{Assert} \to_m \mathcal{P}(\mathit{Heap}). \tag{2}$$

The subscript $-_m$ indicates that we consider only the subset of monotonic functions. By restricting the codomain to subsets of *Heap* that satisfy particular conditions, and by further restricting the function space, we can find a solution to (a variant of) equation (2) in a category of complete metric spaces. However, because monotonicity is defined in terms of the ordering over assertions, which itself is defined in terms of the operation of composition $\circ$ on W (recall that W and *Assert* are isomorphic), we cannot do this using "off-the-shelf" techniques for solving recursive domain equations, like those of Rutten [12] or Birkedal et al. [13]. Instead, we obtain a solution by explicitly constructing the inverse limit of an appropriately chosen sequence of approximations to (2). We discuss these challenges and our solution in more detail in Section 4.

Contributions In summary, our main contributions are the following:

- We highlight which semantic ingredients are critical in establishing the validity of the frame and anti-frame rules. We hope that this will lead to increased understanding, and expect that these ingredients can be used as building blocks in the soundness proofs of future logics with information hiding principles.
- We give a proof of the soundness of a program logic that includes frame and anti-frame rules for higher-order store.

To improve the readability, we omit several proofs in the main text; some are found in the appendices.

2 Semantic setup

In this section, we describe semantic ingredients that can be used to validate (certain types of) anti-frame and frame rules.

Programming language. Our assumptions on the semantics of the programming language are fairly standard. We assume that there is a (pointed, chain-complete partially ordered) set of heaps, *Heap*, and that commands either diverge, terminate successfully, or fault, i.e., that $\mathit{Com} = \mathit{Heap} \multimap (\mathit{Heap} \oplus \{\mathit{error}\}_\perp)$ is the set of (strict continuous) functions into *Heap* with an error element adjoined. We also assume that there exists a family of projection functions $(\pi_k : \mathit{Heap} \multimap \mathit{Heap})_{k\in\mathbb{N}}$.

The images of these projection functions must contain only finite elements[2] and the projection functions must satisfy the following conditions:

- $\bot = \pi_0(h) \sqsubseteq \ldots \sqsubseteq \pi_k(h) \sqsubseteq \pi_{k+1}(h) \sqsubseteq \ldots \sqsubseteq h$ for all $h \in \mathit{Heap}$, i.e., the π_k's form an increasing chain of approximations of the identity on *Heap*;
- $\pi_j \circ \pi_k = \pi_{\min\{j,k\}}$ for all j, k; in particular, every π_k is idempotent;
- $\bigsqcup_k \pi_k(h) = h$, i.e., every heap is the limit of its approximations.

For example, these conditions hold if *Heap* is an SFP domain (e.g., [14]) with a particular choice of projections. Finally, we assume a partial commutative associative operation $h \cdot h'$, which intuitively lets us combine heaps with disjoint locations and is compatible with the projections: $\pi_k(h \cdot h') = \pi_k(h) \cdot \pi_k(h')$.

We write $\mathbb{N}_\infty$ for the natural numbers extended with ∞, with $\infty + k = k + \infty = \infty$. We define $\pi_\infty = id$ and $2^{-\infty} = 0$. The *rank of* h, written $\mathit{rnk}(h)$, is the least $k \in \mathbb{N}_\infty$ such that $\pi_k(h) = h$.

Uniformity and distance. Our program logics concern properties of heaps that are closed under the projection functions π_k. We write *UAdm* for the set of admissible[3] subsets $p \subseteq \mathit{Heap}$ that are *uniform*: for any $k \in \mathbb{N}$, if $h \in p$ then $\pi_k(h) \in p$. Using the heap combination operation, we define separating conjunction $p * q$ for $p, q \in \mathit{UAdm}$ in the usual way: $h \in p * q$ iff $h = h_1 \cdot h_2$ for some $h_1 \in p$ and $h_2 \in q$. We assume that $p * q$ is in *UAdm*.[4]

Uniformity gives rise to a notion of distance between (or, similarity of) properties of heaps. More precisely, writing $\pi_k(p)$ for the image of p under the projection π_k, the function $d(p, q) = 2^{-\sup\{k \in \mathbb{N}_\infty \mid \pi_k(p) = \pi_k(q)\}}$ defines a notion of distance on *UAdm*. This function satisfies the requirements of a *1-bounded ultrametric*: that is, d takes real values in the interval $[0, 1]$, is symmetric, is such that $d(p, q) = 0$ holds iff $p = q$, and satisfies $d(p, q) \leq \max\{d(p, r), d(r, q)\}$ for all $p, q, r \in \mathit{UAdm}$. With respect to this metric, *UAdm* is *complete* in the usual sense that every Cauchy sequence has a limit. The metric and this completeness result of *UAdm* make it possible to model recursively defined assertions using the Banach fixed point theorem.

Worlds and assertions. Our semantics of assertions is defined in the category *CBUlt* of complete 1-bounded ultrametric spaces and non-expansive functions. This means that every semantic domain involved in the semantics has an appropriate notion of distance and that every function is non-expansive, i.e., the distance between two outputs is no greater than the distance between the two corresponding inputs.

The main ingredients necessary for validating forms of anti-frame and frame rules are a set of possible worlds, and an interpretation of the worlds as assertions. Thus, we require:

[2] An element d in a cpo D is finite iff for all chains $\{d_n\}_{n \in \omega}$ in D, $d \sqsubseteq \bigsqcup_{n \in \omega} d_n$ implies that $d \sqsubseteq d_n$ for some n.

[3] The admissibility of p means that p is closed under limits of chains and contains $\bot$.

[4] This assumption holds when *Heap* is constructed in a standard way in terms of finite partial functions or records, just like our model in Sections 4 and 5.

1. A monoid $(W, e, \circ)$ of *worlds,* or *invariants,* where W is an object in *CBUlt* and the operation $\circ$ is non-expansive with respect to this metric. The monoid structure induces a preorder $\sqsubseteq$ on W, by $w \sqsubseteq w' \Leftrightarrow \exists w_0 \in W.\, w' = w \circ w_0$. Note that $\circ$ is in general not commutative, and that w' is obtained by extending w on the *right.* Using this preorder, we define a domain

$$\mathit{Assert} \stackrel{\mathit{def}}{=} (\tfrac{1}{2} \cdot W) \to_m \mathit{UAdm}.$$

   for assertions. Here, $\frac{1}{2} \cdot W$ denotes the scaling of the distance function on W by $1/2$, and the function space consists of the non-expansive *monotone* functions. Assertions are thus parameterized by worlds, and this parameterization satisfies two conditions. The first condition is *contractiveness*, meaning that the distance between worlds gets reduced when they are used in an assertion: $d(p(w_0), p(w_1)) \leq d(w_0, w_1)/2$ for all $p \in \mathit{Assert}$ and all $w_0, w_1 \in W$. In the definition, contractiveness is formalized in two steps, first by scaling down the distance of worlds by $1/2$ and then by stipulating that assertions should preserve this scaled-down distance (i.e., be non-expansive).
   The second condition is monotonicity: $p(w) \subseteq p(w \circ w_0)$ holds for all $p \in \mathit{Assert}$ and all $w, w_0 \in W$. Here, w_0 can be thought of as an invariant that is hidden in world w and revealed in world $w \circ w_0$. If some heap h satisfies the assertion p while w_0 is hidden, then h still satisfies p after w_0 is revealed. Intuitively, because the commands stored in the heap h do not know about the invariant w_0, they must preserve it.
2. A non-expansive *coercion* function $i : W \to \mathit{Assert}$, which offers a way of interpreting worlds as assertions.
   We do not in general require $W \cong \mathit{Assert}$: a one-way coercion is sufficient for our purposes. In fact, it is possible to define instances of our framework where W is strictly "smaller" than *Assert*. Such a restriction, where not every assertion can play the role of a hidden invariant, can be exploited to establish the soundness of stronger versions of anti-frame and frame rules than the ones in the literature [4,9]. For details, see Appendix D.

For the moment, we simply assume that the above ingredients are provided. In Section 4, we actually construct a particular set of worlds, together with an appropriate coercion function from worlds to assertions. In this particular case, $W \cong \mathit{Assert}$ holds.

The parameterization of assertions by a monoid W has the interesting consequence that the following $\otimes$ operator, from $\mathit{Assert} \times W$ to *Assert*:

$$(p \otimes w) \stackrel{\mathit{def}}{=} \lambda w_0.\, p\,(w \circ w_0)$$

is an action of this monoid over assertions, that is, it satisfies $p \otimes e = p$ and $(p \otimes w) \otimes w_0 = p \otimes (w \circ w_0)$. These are the semantic analogues of two of the distribution axioms in Pottier's type system [4], and are also included in the logic of Section 5.

Healthiness conditions. The monoid of worlds and the coercion function must satisfy two further compatibility conditions. To express these, we use the ab-

breviation $p \circ w \stackrel{def}{=} p \otimes w * i(w)$, where $*$ denotes the pointwise lifting of the separating conjunction on *UAdm* to *Assert*. The first condition is:

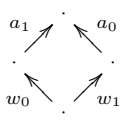
Condition 1. Coercions preserve $- \circ w_0$: $\forall w, w_0 \in W.\ i(w \circ w_0) = i(w) \circ w_0$.

This condition lets us explain the extension of one invariant w by a second invariant w_0 in terms of assertions. By unfolding the definition, we see that $i(w \circ w_0)$ is the assertion obtained by $*$-conjoining the assertion $i(w_0)$ to $i(w)$, and additionally ensuring that all computations described by the latter also preserve the invariant w_0.

The asymmetric nature of Condition 1 indicates that we cannot in general expect the monoid to be commutative. Instead, we require a weaker property: the existence of commutative pairs.

Definition 1 (Commutative pair). *Let w_0, w_1, a_0 and a_1 be worlds. The pair (a_0, a_1) is a* commutative pair for (w_0, w_1) *iff (1) $w_0 \circ a_1 = w_1 \circ a_0$, (2) $i(w_0) \otimes a_1 = i(a_0)$ and (3) $i(w_1) \otimes a_0 = i(a_1)$.*

If (a_0, a_1) is a commutative pair for (w_0, w_1) then $w_0 \circ a_1 = w_1 \circ a_0$ provides an upper bound of w_0 and w_1 with respect to the extension order $\sqsubseteq$. Intuitively, we can "merge" two invariants, ensuring that all computations described in the first invariant preserve the second invariant, and vice versa.



Condition 2. Every pair (w_0, w_1) of worlds has a commutative pair.

Pottier's *revelation lemma* [4], which forms the core of his sketch of a syntactic soundness argument for his anti-frame rule, assumes the existence of commutative pairs. Commutative pairs play a similarly important role in the semantic soundness proofs below. The model described in Section 4 gives a rigorous justification for their existence.

As a consequence of Condition 1 and of the fact that $\otimes$ is a monoid action, we have the following lemma:

Lemma 2. *For all $p \in Assert$ and all $w, w_0 \in W$, $(p \circ w) \circ w_0 = p \circ (w \circ w_0)$.*

3 Semantic triples, anti-frame rule and frame rules

In this section we consider the soundness of specific versions of anti-frame and frame rules, based on the semantic setting described above. For a command $c \in Com$, let $\pi_k(c)$ be the command defined by $\pi_k(c)(h) = error$ if $c(\pi_k(h)) = error$, and by $\pi_k(c)(h) = \pi_k(c(\pi_k(h)))$ if $c(\pi_k(h)) \in Heap$. Note that $\pi_\infty(c) = c$.

Definition 3. *Let tri be the ternary predicate on $Assert \times Com \times Assert$ such that $tri(p, c, q)$ holds iff*

$$\forall u \in UAdm.\, \forall h \in p(e) * u.\ \ c(h) \in \mathrm{Ad}(\textstyle\bigcup_w (q \circ w)(e) * u),$$

where $\mathrm{Ad}(-)$ is the admissible downward closure.

$$\textsc{Anti-frame}\quad \frac{\models\{p \otimes w_0\}c\{q \circ w_0\}}{\models\{p\}c\{q\}} \qquad \textsc{Deep-frame}\quad \frac{\models\{p\}c\{q\}}{\models\{p \circ w_0\}c\{q \circ w_0\}} \qquad \textsc{Shallow-frame}\quad \frac{}{\{p\}c\{q\} \models \{p * i(w_0)\}c\{q * i(w_0)\}}$$

Fig. 1. Semantic versions of a basic form of anti-frame and frame rules

This definition deserves some explanation. First, the universal quantification over $u \in \mathit{UAdm}$ in the definition of $\mathit{tri}(p, c, q)$ "bakes in" the first-order frame rule, i.e., $\mathit{tri}(p, c, q)$ is only true of commands c that validate the first-order frame rule. Next, the existential quantification (union) over worlds $w \in W$ achieves the hiding of state from the post-condition of triples, as expressed by anti-frame rules. Because uniform admissible sets are not closed under arbitrary unions, we take the admissible downward closure. Technically, this makes sense because we assume commands are continuous and because we consider partial correctness only. We view the post-condition $\mathrm{Ad}(\bigcup_w (q \circ w)(e) * u) \subseteq \mathit{Heap}$ as a subset of $\mathit{Heap} \oplus \{\mathit{error}\}_\perp$ in the evident way. In particular, this means that $\mathit{tri}(p, c, q)$ is only true of commands c that do not fault for states in the pre-condition. This definition does not "bake in" monotonicity w.r.t. invariants (worlds): p and $q \circ w$ are "closed" just by applying them to the empty world. This is rectified in the following definition of validity:

Definition 4 (Validity). *A (semantic) triple* $\{p\}c\{q\}$ holds with respect to w and $k \in \mathbb{N}_\infty$, *which we write* $w \models_k \{p\}c\{q\}$, *iff* $\mathit{tri}(p \circ (w \circ w_0), \pi_k(c), q \circ (w \circ w_0))$ *holds for all* $w_0 \in W$. *We sometimes omit the index when* $k = \infty$.

As a consequence of the quantification over worlds w_0 in this definition, the validity of a triple is monotonic: if $w \models_k \{p\}c\{q\}$ and $w \sqsubseteq w'$ then $w' \models_k \{p\}c\{q\}$. The approximate validity (i.e., the case where $k \neq \infty$) is used when considering *nested* triples. Recall that assertions are the *contractive* monotone functions from W to UAdm.[5] Because nested triples will be interpreted as elements in Assert, they must be contractive. The approximations will allow us to satisfy this requirement. We write $\models\{p\}c\{q\}$ to mean that $w \models_k \{p\}c\{q\}$ for all k, w. Also, we write $\{p\}c\{q\} \models \{p'\}c'\{d'\}$ to mean that $w \models_k \{p\}c\{q\} \Rightarrow w \models_k \{p'\}c'\{q'\}$ holds for all w, k.

We are now ready to describe semantic versions of examples of anti-frame and frame rules and to prove their soundness. The semantic rules are given in Fig. 1, where the first two rules should be understood as the implication from the premise to the conclusion.

Our first lemma is a consequence of the monotonicity of assertions.

Lemma 5. *For all* w, *we have that (1)* $\mathit{tri}(p \otimes w, c, q) \Rightarrow \mathit{tri}(p, c, q)$ *and (2)* $\mathit{tri}(p, c, q \circ w) \Rightarrow \mathit{tri}(p, c, q)$.

[5] More precisely, they are the non-expansive monotone functions from $\frac{1}{2} \cdot W$ to UAdm.

Proof. For the first implication, suppose that $u \in \mathit{UAdm}$ and $h \in p(e) * u$. By the monotonicity of p and by $e \sqsubseteq w$ we obtain $p(e) \subseteq p(w) = p(w \circ e) = (p \otimes w)(e)$. Therefore, $h \in (p \otimes w)(e) * u$. The result now follows from the assumption that $\mathit{tri}(p \otimes w, c, q)$ holds.

For the second implication, suppose again that $u \in \mathit{UAdm}$ and $h \in p(e) * u$. We must show that $c(h) \in \mathrm{Ad}(\bigcup_{w'}(q \circ w')(e) * u)$. From the assumption that $\mathit{tri}(p, c, q \circ w)$ holds we obtain $c(h) \in \mathrm{Ad}(\bigcup_{w''}((q \circ w) \circ w'')(e) * u)$. By Lemma 2,

$$\textstyle\bigcup_{w''}((q \circ w) \circ w'')(e) \;=\; \bigcup_{w''}(q \circ (w \circ w''))(e) \;\subseteq\; \bigcup_{w'}(q \circ w')(e).$$

The result then follows from the monotonicity of $*$ and the monotonicity of the closure operation $\mathrm{Ad}(\cdot)$. □

The next lemma amounts to gluing two commutative pair diagrams together (along a_0 there). Its proof involves the associativity of $\circ$ as well as Condition 1.

Lemma 6. *If (a_0, a_1) is a commutative pair for (w_0, w_1) and (b_0, a_2) is a commutative pair for (a_0, w_2) then $(b_0, a_1 \circ a_2)$ is a commutative pair for $(w_0, w_1 \circ w_2)$.*

The following proposition combines Lemmas 5 and 6, and relates the validity of two triples in our anti-frame rule.

Proposition 7. *For all worlds (w_0, w), if (a_0, a) is a commutative pair for (w_0, w), then $a \models_k \{p \otimes w_0\}c\{q \circ w_0\}$ implies $w \models_k \{p\}c\{q\}$.*

Proof. We need to show that $w \models_k \{p\}c\{q\}$, which by definition means that for all w_1, letting $w_2 = w \circ w_1$ and $d = \pi_k(c)$,

$$\mathit{tri}(p \circ w_2,\ d,\ q \circ w_2). \tag{3}$$

By Condition 2, there exists a commutative pair (b_0, a_1) for (a_0, w_1). Let $b_2 = a \circ a_1$. By Lemma 6, (b_0, b_2) is a commutative pair for $(w_0, w \circ w_1) = (w_0, w_2)$. In particular, we have $w_0 \circ b_2 = w_2 \circ b_0$ and $i(b_2) = i(w_2) \otimes b_0$. The assumed triple implies $\mathit{tri}((p \otimes w_0) \circ b_2,\ d,\ (q \circ w_0) \circ b_2)$. Thus,

$$\mathit{tri}((p \circ w_2) \otimes b_0,\ d,\ (q \circ w_2) \circ b_0) \tag{4}$$

follows, using the following equalities:

$$\begin{aligned}(p \otimes w_0) \circ b_2 \;&=\; p \otimes (w_0 \circ b_2) * i(b_2) \;=\; p \otimes (w_2 \circ b_0) * i(w_2) \otimes b_0 \\ &=\; (p \otimes w_2) \otimes b_0 * i(w_2) \otimes b_0 \;=\; (p \otimes w_2 * i(w_2)) \otimes b_0 \\ &=\; (p \circ w_2) \otimes b_0, \\ (q \circ w_0) \circ b_2 \;&=\; q \circ (w_0 \circ b_2) \;=\; q \circ (w_2 \circ b_0) \;=\; (q \circ w_2) \circ b_0.\end{aligned}$$

From (4), we derive the desired triple (3) as shown below:

$$\begin{aligned}\mathit{tri}((p \circ w_2) \otimes b_0, d, (q \circ w_2) \circ b_0) &\Rightarrow \mathit{tri}(p \circ w_2, d, (q \circ w_2) \circ b_0) \\ &\Rightarrow \mathit{tri}(p \circ w_2, d, q \circ w_2).\end{aligned}$$

Both implications hold because of Lemma 5. □

Corollary 8 (Anti-frame rule). *The anti-frame rule in Fig. 1 is sound.*

Proof. Pick w, k. Let p, c, q, w_0 be as in the anti-frame rule in Fig. 1. We must prove that $w \models_k \{p\}c\{q\}$. By Condition 2, there exists a commutative pair (a_0, a) for (w_0, w). By assumption, we have $\models \{p \otimes w_0\}c\{q \circ w_0\}$, so, in particular, $a \models_k \{p \otimes w_0\}c\{q \circ w_0\}$. By Proposition 7, this implies $w \models_k \{p\}c\{q\}$, as desired. □

Next, we move on to the soundness proof of the two frame rules in Fig. 1.

Lemma 9. *The following equivalence, which expresses a distribution axiom [9], holds:* $w_0 \circ w \models_k \{p\}c\{q\}$ *iff* $w \models_k \{p \circ w_0\}c\{q \circ w_0\}$.

Proof. Pick w_1. Let $w_1' = w \circ w_1$. By Definition 4 and the associativity of $\circ$, it suffices to prove the equivalence of $tri(p \circ (w_0 \circ w_1'), \pi_k(c), q \circ (w_0 \circ w_1'))$ and $tri((p \circ w_0) \circ w_1', \pi_k(c), (q \circ w_0) \circ w_1')$. This equivalence follows from Lemma 2. □

Corollary 10 (Frame rules). *The frame rules in Fig. 1 are sound.*

Proof. The soundness of the deep frame rule follows from Lemma 9. The shallow rule is sound thanks to the universal quantification over u in Definition 3. □

4 A concrete model with recursively defined worlds

In this section, we consider a concrete instance of the general framework described in Sections 2 and 3 where W is isomorphic to $Assert$. The $Assert \to W$ direction of this isomorphism means that all assertions can be used as hidden invariants. This lets us define a semantic model of the program logic that is presented next (Section 5). The heap model in this particular case is given by the following recursively defined cpos:

$$Heap = Rec(Val) \quad Val = Int_\perp \oplus Com_\perp \quad Com = Heap \multimap Heap \oplus \{error\}_\perp \quad (5)$$

where $Rec(Val)$ denotes records with entries in Val labelled by positive natural numbers.[6] These labels serve as addresses or locations. The partial operation $h \cdot h'$ combines two heaps h and h' (i.e., takes the union) whenever the domains of h and h' are disjoint. When $h = \perp$ or $h' = \perp$, $h \cdot h'$ is $\perp$. The empty record provides a unit for heap combination, thus there is also a unit for the separating conjunction on $UAdm$. Finally, the solution of (5) in the category $\mathbf{Cppo}_\perp$ of pointed cpos and strict continuous functions comes equipped with a family of projections π_k that satisfy the requirements of Section 2.

The key result of this section is the following theorem:

Theorem 11. *There exists a monoid* $(W, e, \circ)$*, where* W *is an object in CBUlt with an isomorphism* ι *from* W *to* $(\frac{1}{2} \cdot W) \to_m UAdm$*. The operation* $\circ$ *satisfies*

$$\forall\, w_1, w_2, w \in W.\ \ \iota(w_1 \circ w_2)(w) \ = \ \iota(w_1)(w_2 \circ w) * \iota(w_2)(w).$$

[6] Formally, $Rec(D) = (\Sigma_{N \subseteq_{fin} Nats^+}(N \to D_\downarrow))_\perp$ where $N \to D_\downarrow$ is the cpo of maps from the finite address set N to $D_\downarrow = D - \{\perp\}$ of non-bottom elements of D.

The equation in this theorem is just Condition 1, where the coercion i is taken to be the isomorphism ι.

Construction of the worlds W in Theorem 11. In previous work [9] we gave a model of a separation logic with nested triples and higher-order frame rules, but no anti-frame rule. For this model we needed a solution W' to the following domain equation:[7]

$$W' \;\cong\; (\tfrac{1}{2} \cdot W') \to \mathit{UAdm}. \tag{6}$$

Note that (6) is almost the same domain equation as described in Theorem 11 above, except that there is no restriction to monotonic functions in the function space on the right. One can use a general existence theorem [12,13] to obtain a solution W' for (6) in the category *CBUlt*. In a second step, using the complete metric on W', one can then define a monoid operation $\circ$ that satisfies the equation stated in Theorem 11.

In the present setup, this two-step approach to constructing a solution $W \cong \mathit{Assert}$ and a monoid operation $\circ$ cannot be applied, however, because of the added monotonicity requirement in Theorem 11: since the order on W is defined in terms of the operation $\circ$, one needs $\circ$ already in order to *express* this equation, i.e., it appears necessary to define the operation $\circ$ *at the same time* as W. Thus we construct $W \cong (\frac{1}{2} \cdot W) \to_m \mathit{UAdm}$ explicitly, as (inverse) limit

$$W = \left\{ x \in \textstyle\prod_{k\geq 0} W_k \mid \forall k \geq 0.\ \iota_k^\circ(x_{k+1}) = x_k \right\}$$

of a sequence of "approximations" W_k of W,

$$W_0 \underset{\iota_0^\circ}{\overset{\iota_0}{\rightleftarrows}} W_1 \underset{\iota_1^\circ}{\overset{\iota_1}{\rightleftarrows}} W_2 \underset{\iota_2^\circ}{\overset{\iota_2}{\rightleftarrows}} \cdots \underset{\iota_k^\circ}{\overset{\iota_k}{\rightleftarrows}} W_{k+1} \underset{\iota_{k+1}^\circ}{\overset{\iota_{k+1}}{\rightleftarrows}} \cdots \tag{7}$$

Each W_k is a complete 1-bounded ultrametric space equipped with a non-expansive operation $\circ_k : W_k \times W_k \to W_k$ and a preorder $\sqsubseteq_k$, so that $W_{k+1} = (\frac{1}{2} \cdot W_k) \to_m \mathit{UAdm}$ are the non-expansive and monotone functions with respect to $\sqsubseteq_k$. The maps ι_k and ι_k° are given by $\iota_{k+1}(w) = \pi_{k+2} \circ w \circ \iota_k^\circ$ and $\iota_{k+1}^\circ(w) = \pi_{k+1} \circ w \circ \iota_k$. The diagram (7) forms a *Cauchy tower* [13], in that $\sup_w d_{k+1}(w, (\iota_k \circ \iota_k^\circ)(w))$ and $\sup_w d_k(w, (\iota_k^\circ \circ \iota_k)(w))$ become arbitrarily small as k increases. The operation $\circ_{k+1}$ is defined in terms of $\circ_k$ and ι_k°:

$$(w_1 \circ_{k+1} w_2)(w) \;\stackrel{def}{=}\; w_1\,(\iota_k^\circ(w_2) \circ_k w) * w_2(w).$$

One technical inconvenience is that the $\circ_k$'s are not associative. However, associativity holds "up to approximation k," $d_k((x \circ_k y) \circ_k z,\ x \circ_k (y \circ_k z)) \leq 2^{-k}$, which yields associativity "in the limit" and thus a monoid structure on W by

$$(x_k)_{k\geq 0} \circ (y_k)_{k\geq 0} \stackrel{def}{=} \left(\lim_{j>k} \iota_k^\circ(\ldots(\iota_{j-1}^\circ(x_j \circ_j y_j)))\right)_{k\geq 0}.$$

[7] Technically, we solved a different equation $W' \cong \frac{1}{2}(W' \to \mathit{UAdm})$. This difference is insignificant, since the solution of one equation leads to that of the other equation.

To finish the proof of Theorem 11 one shows that $\iota(w) \stackrel{def}{=} \lim_k(\lambda w'.\, w_{k+1}(w'_k))$ establishes an isomorphism ι between W and $(\frac{1}{2}\cdot W) \to_m UAdm$, which satisfies $\iota(w_1 \circ w_2) = \iota(w_1) \otimes w_2 * \iota(w_2)$ for $(p \otimes w) = \lambda w'.p(w \circ w')$.

The details of the proof are given in the appendix.

Existence of commutative pairs. To show that W forms an instance of our semantic framework, we also need to prove the existence of commutative pairs. Given a pair (w_0, w_1) of worlds, we construct a commutative pair using properties of $\otimes$ and the coercion ι. Since the monoid operation $\otimes$ is contractive in its second argument, so is the function $f(a_0,\, a_1) = \big(\iota^{-1}(\iota(w_0) \otimes a_1),\ \iota^{-1}(\iota(w_1) \otimes a_0)\big)$ on $W \times W$. By the Banach fixed point theorem, there exists a unique pair $(a_0, a_1) = f(a_0, a_1)$. Thus $\iota(a_0) = \iota(w_0) \otimes a_1$ and $\iota(a_1) = \iota(w_1) \otimes a_0$. Since ι is injective, we can prove the remaining $w_0 \circ a_1 = w_1 \circ a_0$ as follows. For all $w \in W$,

$$\begin{aligned}
\iota(w_0 \circ a_1)(w) &= (\iota(w_0) \otimes a_1)(w) * \iota(a_1)(w) && \text{(by Theorem 11 and def. of } \otimes) \\
&= \iota(a_0)(w) * (\iota(w_1) \otimes a_0)(w) && \text{(by the above properties of } a_0,\, a_1) \\
&= \iota(a_0)(w) * \iota(w_1)(a_0 \circ w) && \text{(by def. of } \otimes) \\
&= \iota(w_1 \circ a_0)(w) && \text{(by Theorem 11).}
\end{aligned}$$

Theorem 12. *The monoid $(W, e, \circ)$ in Theorem 11 and the isomorphism $\iota : W \to$ Assert form an instance of the framework in Section 2.*

5 Program logic

We now give one application of our semantic development. We present a program logic for higher-order store, which includes anti-frame and frame rules. Using the results of Sections 3 and 4, we define the semantics of the logic and prove its soundness.

Programming language. Fig. 2 gives the syntax of a small imperative programming language equipped with operations for stored code and heap manipulation. The expressions in the language are integer expressions, variables, and the quote expression 'C' for representing an unevaluated command C. The integer or code value denoted by expression e_1 is stored in a heap cell e_0 using $[e_0]{:=}e_1$, and this stored value is later looked up and bound to the variable y by $\mathsf{let}\ y{=}[e_0]\ \mathsf{in}\ D$. In the case that the value stored in cell e_0 is code 'C', we can run (or "evaluate") this code by executing $\mathsf{eval}\,[e_0]$. As in ML, all variables x, y, z are *immutable*. The language does not include while loops: they can be expressed by stored code (using Landin's knot). The interpretation of commands in the cpo *Com* of (5) is straightforward [9]. The interpretation of the quote operation, 'C', uses the injection of *Com* into *Val* in Section 4.

Assertions and distribution axioms. As in previous work [9], our assertion language is first-order intuitionistic logic, extended with the separating connectives $\mathsf{e}, *$, and the points-to predicate $\mapsto$ [5]. The syntax of assertions appears in Fig. 2. The standard connectives are omitted.

The most distinguishing features of the assertion language are Hoare triples $\{P\}e\{Q\}$ and invariant extensions $P \otimes Q$. The fact that a triple is an assertion

$$
\begin{array}{rll}
e \in \mathit{Exp} ::= & -1 \mid 1 \mid e_1{+}e_2 \mid \ldots \mid x \mid \text{`}C\text{'} & \text{integer expression, variable, quote} \\
C \in \mathit{Com} ::= & [e_1]{:=}e_2 \mid \mathsf{let}\ y{=}[e]\ \mathsf{in}\ C \mid \mathsf{eval}\,[e] & \text{assignment, lookup, unquote} \\
\mid & \mathsf{let}\ x{=}\mathsf{new}\ (e_1,\ldots,e_n)\ \mathsf{in}\ C \mid \mathsf{free}\ e & \text{allocation, disposal} \\
\mid & \mathsf{skip} \mid C_1;C_2 \mid \mathsf{if}\ (e_1{=}e_2)\ C_1\ C_2 & \text{no op, sequencing, conditional} \\
P, Q \in \mathit{Assn} ::= & e_1 \mapsto e_2 \mid \mathsf{e} \mid P * Q & \text{separating connectives} \\
\mid & \{P\}e\{Q\} \mid P \otimes Q & \text{Hoare triple, invariant extension} \\
\mid & X \mid \mu X.P \mid \ldots & \text{assertion variable, recursion}
\end{array}
$$

Fig. 2. Syntax of expressions, commands and assertions

$$
\begin{array}{rcll}
\{P\}e\{Q\} \otimes R & \Leftrightarrow & \{P \circ R\}e\{Q \circ R\} & \\
(\kappa x.P) \otimes R & \Leftrightarrow & \kappa x.(P \otimes R) & (\kappa \in \{\forall, \exists\}, x \notin \mathit{fv}(R)) \\
(P \otimes R) \otimes R' & \Leftrightarrow & P \otimes (R \circ R') & \\
(P \oplus Q) \otimes R & \Leftrightarrow & (P \otimes R) \oplus (Q \otimes R) & (\oplus \in \{\Rightarrow, \wedge, \vee, *\}) \\
P \otimes R & \Leftrightarrow & P & (R \text{ is } \mathsf{e} \text{ or } P \text{ is one of } \mathsf{true}, \mathsf{false}, \mathsf{e}, e \mapsto e', \ldots) \\
(\mu X.P) \otimes R & \Leftrightarrow & \mu X.(P \otimes R) & (X \notin \mathit{fv}(R))
\end{array}
$$

Fig. 3. Axioms for distributing $- \otimes R$

means that triples can be nested. Intuitively, the assertion $P \otimes Q$ denotes a version of P where every (possibly deeply nested) triple receives a copy of Q as an extra $*$-conjunct in its pre- and post-conditions. More precisely, the behaviour of the $\otimes$ operator is described by the axioms in Fig. 3, which let us distribute $\otimes$ through all the constructs of the assertion language. These axioms use the abbreviation $Q \circ R$ for $(Q \otimes R) * R$.

Assertions include assertion variables $X \in \mathcal{X}$, and can be recursively defined: the construct $\mu X.P$ binds X in P and satisfies the axiom $\mu X.P \Leftrightarrow P[X := \mu X.P]$. Not every recursive assertion is permitted: for $\mu X.P$ to be well-formed, we require that P be *formally contractive in* X. In short, this means that every free occurrence of X within P must lie either within a triple or within the second argument of a $\otimes$ construct. (We omit the straightforward inductive definition of formal contractiveness.) Semantically, this requirement ensures that $\mu X.P$ is well-defined as the unique fixed point of P, viewed as a function of X. In particular, all assertions of the form $\mu X.P \otimes X$, where X does not appear in P, are formally contractive. Pottier's applications of the anti-frame rule [4] make extensive use of assertions of this form.

The interpretation of assertions uses W in Section 4. Given such W and an environment η that maps variables x to values $\eta(x) \in \mathit{Val}$, we interpret an assertion P as a non-expansive function $[\![P]\!]_\eta : \mathit{Assert}^{\mathcal{X}} \to \mathit{Assert}$. The uniform admissible sets in UAdm, partially ordered by inclusion, form a complete Heyting algebra with a monotone commutative monoid. The domain Assert, ordered pointwise, inherits this structure (see Appendix B). This is used to interpret the intuitionistic first-order fragment, $*$ and e of the assertion language.

$$[\![P \otimes R]\!]_{\eta,\xi} = [\![P]\!]_{\eta,\xi} \otimes \iota^{-1}([\![R]\!]_{\eta,\xi}) \qquad [\![\mu X.P]\!]_{\eta,\xi} = \mathit{fix}(\lambda q.\ [\![P]\!]_{\eta,\xi[X:=q]})$$
$$[\![\{P\}\text{`}C\text{'}\{Q\}]\!]_{\eta,\xi} = \lambda w.\, \{h \mid \mathit{rnk}(h) > 0 \ \Rightarrow\ w \models_{\mathit{rnk}(h)-1} \{[\![P]\!]_{\eta,\xi}\}\, [\![C]\!]_{\eta}\, \{[\![Q]\!]_{\eta,\xi}\}\}$$

Fig. 4. Interpretation of assertions

$$\textsc{Anti-frame}\ \frac{\Gamma;\Xi \vdash \{P \otimes R\}e\{Q \circ R\}}{\Gamma;\Xi \vdash \{P\}e\{Q\}} \qquad \textsc{Deep-frame}\ \frac{\Gamma;\Xi \vdash \{P\}e\{Q\}}{\Gamma;\Xi \vdash \{P \circ R\}e\{Q \circ R\}} \qquad \textsc{Shallow-frame}\ \frac{}{\Gamma;\Xi \vdash \{P\}e\{Q\} \Rightarrow \{P * R\}e\{Q * R\}}$$

Fig. 5. Proof rules from separation logic

Fig. 4 shows three of the remaining cases. First, via the isomorphism ι^{-1}, we can turn any assertion $r \in \mathit{Assert}$ into an invariant $\iota^{-1}(r) \in W$ and thus interpret the invariant extension $P \otimes R$. Next, because P must be formally contractive in X, the map $q \mapsto [\![P]\!]_{\eta,\xi[X:=q]}$ on Assert is contractive in the metric sense: thus, by the Banach fixed point theorem, it has a unique fixed point. Finally, the interpretation of nested triples is in terms of semantic triples, and uses approximate validity to ensure non-expansiveness.

Proof rules. The logic derives judgements of the form $\Gamma;\Xi \vdash P$, where P is an assertion, and Γ and Ξ respectively bind variables and assertion variables. For instance, to prove that command C stores at cell 1 some code that writes 0 into cell 10, one would need to derive $\Gamma;\Xi \vdash \{1 \mapsto \_\}\text{`}C\text{'}\{1 \mapsto \{10 \mapsto \_\}\_\{10 \mapsto 0\}\}$.

The logic includes the standard proof rules for intuitionistic logic and the logic of bunched implications [15] as well as standard separation logic proof rules [9]. We do not repeat these rules here. Fig. 5 shows a version of the anti-frame rule and two versions of the frame rule: the deep frame rule (expressed in combination with the distribution axioms) and the first-order shallow frame rule (which takes the form of an axiom).

Theorem 13 (Soundness). *The interpretation of assertions is well-defined, and validates the distribution axioms of Fig. 3 and the inference rules of Fig. 5.*

6 Conclusion and Future Work

We have presented a semantic framework for studying the soundness of anti-frame and frame rules for languages with higher-order store. Moreover, we have presented a concrete instance of the framework, and used it to give the first rigorous proof of soundness of separation logic with anti-frame and frame rules for a language with higher-order store.

We are aware of other instantiations of the semantic framework, which can be used to show the soundness of stronger variants of the anti-frame and frame

rules, provided the universe W of invariants is restricted. See Appendix D for details.

Future work includes lifting the results in this paper to Pottier's type-and-capability system as well as extending our soundness results to generalized versions of the anti-frame and frame rules where invariants evolve in more sophisticated ways over time [16,17].

Acknowledgments We would like to thank Kristian Støvring and Jacob Thamsborg for helpful discussions.

Partial support has been provided by FNU project 272-07-0305 "Modular reasoning about software", and EPSRC projects EP/G003173/1 "From reasoning principles for function pointers to logics for self-configuring programs" and EP/E053041/1 "Scalable program analysis for software verification".

References

1. Parkinson, M., Bierman, G.: Separation logic and abstraction. In: POPL. (2005) 247–258
2. Biering, B., Birkedal, L., Torp-Smith, N.: BI-hyperdoctrines, higher-order separation logic, and abstraction. TOPLAS **29**(5) (2007)
3. Parkinson, M., Bierman, G.: Separation logic, abstraction and inheritance. In: POPL. (2008) 75–86
4. Pottier, F.: Hiding local state in direct style: a higher-order anti-frame rule. In: LICS. (2008) 331–340
5. Reynolds, J.C.: Separation logic: A logic for shared mutable data structures. In: LICS. (2002) 55–74
6. O'Hearn, P.W., Yang, H., Reynolds, J.C.: Separation and information hiding. In: POPL. (2004) 268–280
7. Birkedal, L., Torp-Smith, N., Yang, H.: Semantics of separation-logic typing and higher-order frame rules for Algol-like languages. LMCS **2**(5:1) (2006)
8. Birkedal, L., Reus, B., Schwinghammer, J., Yang, H.: A simple model of separation logic for higher-order store. In: ICALP. (2008) 348–360
9. Schwinghammer, J., Birkedal, L., Reus, B., Yang, H.: Nested Hoare triples and frame rules for higher-order store. In: CSL. (2009) 440–454
10. Pottier, F.: Three comments on the anti-frame rule. Unpublished note (July 2009)
11. Levy, P.B.: Possible world semantics for general storage in call-by-value. In: CSL. (2002) 232–246
12. Rutten, J.J.M.M.: Elements of generalized ultrametric domain theory. TCS **170**(1–2) (December 1996) 349–381
13. Birkedal, L., Støvring, K., Thamsborg, J.: The category-theoretic solution of recursive metric-space quations. Technical Report ITU-2009-119, IT University of Copenhagen (2009)
14. Streicher, T.: Domain-theoretic Foundations of Functional Programming. World Scientific (2006)
15. O'Hearn, P.W., Pym, D.J.: The logic of bunched implications. Bulletin of Symbolic Logic **5**(2) (June 1999) 215–244
16. Pilkiewicz, A., Pottier, F.: The essence of monotonic state. Submitted (October 2009)
17. Pottier, F.: Generalizing the higher-order frame and anti-frame rules. Unpublished note (July 2009)

A Composition of commutative pairs

This appendix contains the proof of Lemma 6, which describes how commutative pairs compose. We repeat the lemma below:

Lemma 6 If (a_0, a_1) is a commutative pair for (w_0, w_1) and if (b_0, a_2) is also a commutative pair for (a_0, w_2), then $(b_0, a_1 \circ a_2)$ is a commutative pair for $(w_0, w_1 \circ w_2)$.

Proof. We prove the three defining equations for $(b_0, a_1 \circ a_2)$ being a commutative pair for $(w_0, w_1 \circ w_2)$ below. First,

$$\begin{aligned} w_0 \circ (a_1 \circ a_2) &= (w_0 \circ a_1) \circ a_2 = (w_1 \circ a_0) \circ a_2 \\ &= w_1 \circ (a_0 \circ a_2) = w_1 \circ (w_2 \circ b_0) \\ &= (w_1 \circ w_2) \circ b_0. \end{aligned}$$

The most interesting equalities here are the second and the fourth, which follow from the assumptions on commutative pairs made in the lemma. The other equalities are by the associativity of the monoid operation. Next, we prove the equations for $i(w_0)$:

$$\begin{aligned} i(w_0) \otimes (a_1 \circ a_2) &= (i(w_0) \otimes a_1) \otimes a_2 \\ &= i(a_0) \otimes a_2 = i(b_0). \end{aligned}$$

The second and third equalities are important, and they follow from the assumptions on commutative pairs. Finally, we prove the remaining equation for $i(w_1 \circ w_2)$:

$$\begin{aligned} i(w_1 \circ w_2) \otimes b_0 &= (i(w_1) \otimes w_2 * i(w_2)) \otimes b_0 \\ &= (i(w_1) \otimes w_2 \otimes b_0) * (i(w_2) \otimes b_0) \\ &= (i(w_1) \otimes (w_2 \circ b_0)) * i(a_2) \\ &= (i(w_1) \otimes (a_0 \circ a_2)) * i(a_2) \\ &= (i(w_1) \otimes a_0 \otimes a_2) * i(a_2) \\ &= (i(a_1) \otimes a_2) * i(a_2) = i(a_1 \circ a_2). \end{aligned}$$

The third, fourth and sixth equalities use the assumptions on commutative pairs, and the last equality follows from Condition 1. □

B Complete Heyting algebra and monotone commutative monoid of assertions in Sections 4 and 5

This technical appendix contains proofs about the complete Heyting algebra structure and the monotone commutative monoid on *Assert* in Sections 4 and 5 that have been omitted from the main part of the paper.

Lemma 14 (Separating conjunction). *Separating conjunction is well-defined: if $p, q \in UAdm$ then so is $p * q$.*

Proof. We first show that separating conjunction on *UAdm* is well-defined, i.e., if $p, q \in UAdm$ then so is $p * q$. Since $\bot \in p$ and $\bot \in q$, and $\bot = \bot \cdot \bot$, we have $\bot \in p * q$. If $h_0 \sqsubseteq h_1 \sqsubseteq \ldots$ is a chain in $p * q$ with lub $h \neq \bot$, then $h_i \neq \bot$ for almost all i and there are heaps h'_i, h''_i such that $h_i = h'_i \cdot h''_i$ and $h'_i \in p$ and $h''_i \in q$. Since the order on *Heap* is defined pointwise (for heaps with equal domain) and the domain of each heap is finite, there must be a subsequence $(h_{i_k})_k$ such that $h'_{i_1} \sqsubseteq h'_{i_2} \sqsubseteq \ldots$ is a chain in p and $h''_{i_1} \sqsubseteq h''_{i_2} \sqsubseteq \ldots$ is a chain in q. Thus, also their lubs h' and h'' satisfy $h' \in p$ and $h'' \in q$. It follows that $h = h' \cdot h'' \in p * q$, and therefore $p * q$ is admissible. To see that $p * q$ is uniform, suppose $h \in p * q$ and let $n \in \omega$. By definition, $h = h_1 \cdot h_2$ for heaps h_1, h_2 such that $h_1 \in p$ and $h_2 \in q$. By uniformity of p and q, this gives $\pi_n(h_1) \in p$ and $\pi_n(h_2) \in q$, from which $\pi_n(h) = \pi_n(h_1) \cdot \pi_n(h_2) \in p * q$ follows. □

Lemma 15 (Complete Heyting algebra and monotone commutative monoid structures on *UAdm*). *Let $\mathsf{e} = \{\{\!|\}\!\}, \bot\}$ where the "empty heap" $\{\!|\}\!\}$ denotes the unit for the operation $h \cdot h'$. Then $(UAdm, \subseteq)$ is a complete Heyting algebra with a monotone commutative monoid structure $(UAdm, *, \mathsf{e})$, where the monotonicity means that the $*$ operator is monotone.*

Proof. Since admissibility and uniformity are preserved by arbitrary intersections, *UAdm* is a complete lattice, with meets given by set-theoretic intersection, least element $\{\bot\}$ and greatest element *Heap*. Binary joins are given by set-theoretic union, and arbitrary joins by $\bigsqcup_i p_i = \bigcap\{p \in UAdm \mid p \supseteq \bigcup_i p_i\}$.

The join is described more explicitly as $\bigsqcup_i p_i = \{h \mid \forall n \in \omega.\ \pi_n(h) \in \bigcup_i p_i\}$. First, note that the right hand side $r \stackrel{def}{=} \{h \mid \forall n \in \omega.\ \pi_n(h) \in \bigcup_i p_i\}$ is an element of *UAdm*: r is uniform, i.e., $h \in r$ implies $\pi_m(h) \in r$ for all $m \in \omega$, since $\pi_n \cdot \pi_m = \pi_{\min\{n,m\}}$. To show that r is also admissible suppose $h_0 \sqsubseteq h_1 \sqsubseteq \ldots$ is a chain in r, and let h be the lub of this chain. We must show that $\pi_n(h) \in \bigcup_i p_i$ for all $n \in \omega$. By compactness, $\pi_n(h) \sqsubseteq h_k \sqsubseteq h$ for some k, and hence $\pi_n(h) = \pi_n(h_k) \in \bigcup p_i$ using the idempotency of π_n and the fact that $h_k \in r$. To see the inclusion $r \subseteq \bigsqcup_i p_i$, note that for all h, if $\pi_n(h) \in \bigcup_i p_i \subseteq p$ for all $n \in \omega$ and some arbitrary $p \in UAdm$, then also $h = \sqcup_n \pi_n(h) \in p$ by admissibility, and hence $h \in \bigsqcup_i p_i$ follows. For the other inclusion, we claim that the right hand side $r \stackrel{def}{=} \{h \mid \forall n \in \omega.\ \pi_n(h) \in \bigcup_i p_i\}$ is one of the elements appearing in the intersection; from this claim it is immediate that $r \supseteq \bigsqcup_i p_i$. The claim follows since $r \supseteq \bigcup_i p_i$ by the uniformity of the p_i's.

The implication of this complete lattice *UAdm* is described by $p \Rightarrow q \stackrel{def}{=} \{h \mid \forall n \in \omega.\ \text{if } \pi_n(h) \in p \text{ then } \pi_n(h) \in q\}$: Using $\pi_n \cdot \pi_m = \pi_{\min\{n,m\}}$ it is easy to see that $p \Rightarrow q$ is uniform. Admissibility follows analogously to the case of joins: if $h_0 \sqsubseteq h_1 \sqsubseteq \ldots$ is a chain in $p \Rightarrow q$ with lub h, and if $n \in \omega$ is such that $\pi_n(h) \in p$ then we must show that $\pi_n(h) \in q$. Since $\pi_n(h) \sqsubseteq h$ is compact, there is some k such that $\pi_n(h) \sqsubseteq h_k \sqsubseteq h$, and thus the required $\pi_n(h) = \pi_n(h_k) \in q$ follows from $h_k \in p \Rightarrow q$. Next, to see that $p \Rightarrow q$ is indeed the implication

in $UAdm$, first note that we have $p \cap (p \Rightarrow q) \subseteq q$, using the uniformity of p and the admissibility of q. If $p \cap r \subseteq q$ for some $r \in UAdm$, and $h \in r$ and $\pi_n(h) \in p$ for some $n \in \omega$, then the uniformity of r yields $\pi_n(h) \in q$. Thus we obtain $p \cap r \subseteq q \Leftrightarrow r \subseteq p \Rightarrow q$.

That $*$ is an operation on $UAdm$ is established in Lemma 14. It is easy to check that $*$ is commutative and associative and that it is monotone, i.e., if $p \subseteq p'$ and $q \subseteq q'$ then $p * q \subseteq p' * q'$. Moreover, we have $\mathsf{e} \in UAdm$, and the fact that $p * \mathsf{e} = p = \mathsf{e} * p$ follows from the definition of the heap combination $h \cdot h'$. □

Lemma 16 (Completeness). *Let A be a complete 1-bounded ultrametric space and $\sqsubseteq$ be a preorder on A. The set of non-expansive monotone functions from A to $UAdm$*

$$A \to_m UAdm \stackrel{def}{=} \{f : A \to UAdm \mid f \text{ is non-expansive, and } \forall w, w' \in A.\ w \sqsubseteq w' \implies f(w) \subseteq f(w')\},$$

is a complete 1-bounded ultrametric space, when it is equipped with the sup-*metric.*

Proof. Recall that the non-expansive functions from A to $UAdm$ with the sup metric forms a complete 1-bounded ultrametric space. Let $A \to UAdm$ be this ultrametric space. Since $A \to_m UAdm$ is a subset of $A \to UAdm$, it is also a 1-bounded ultrametric space. Thus, it is sufficient to show the completeness of $A \to_m UAdm$. To this end, consider a Cauchy sequence $\{f_n\}_n$ in $A \to_m UAdm$. Then, $\{f_n\}_n$ is also a Cauchy sequence in $A \to UAdm$. Thus, it has the limit f there, because $A \to UAdm$ is complete. Furthermore, this limit is defined pointwise. We will prove that f is in $A \to_m UAdm$. Let w, w' be in A such that $w \sqsubseteq w'$. Our goal now is to prove that $f(w) \subseteq f(w')$.

As a first step of our proof, we simplify the goal as follows, exploiting the admissibility of $f(w')$:

$$\forall k \in \omega.\ \pi_k(f(w)) \ \subseteq\ f(w'). \tag{8}$$

To see why this implies the goal, choose h from $f(w)$. Then, $\pi_k(h) \in \pi_k(f(w))$ for all k. Thus, (8) implies that $\pi_k(h)$ is also in $f(w')$ for all k. Furthermore, $\{\pi_k(h)\}_k$ is a chain with the limit h, so the admissibility of $f(w')$ means that h has to be in $f(w')$ as well.

Next, we prove (8). For this, consider an index k in ω. Since f is the limit of a Cauchy sequence $\{f_n\}_n$, there exists m such that for all $n \geq m$, $d(f_n, f) \leq 2^{-k}$ in $A \to UAdm$. In particular, this means that for the same m,

$$\pi_k(f_m(w)) \ =\ \pi_k(f(w)) \qquad \text{and} \qquad \pi_k(f_m(w')) \ =\ \pi_k(f(w')) \ \subseteq\ f(w'). \tag{9}$$

Now, note that $f_m(w) \subseteq f_m(w')$ by the monotonicity of f_m, so that $\pi_k(f_m(w)) \subseteq \pi_k(f_m(w'))$. This and (9) imply that so $\pi_k(f(w)) \subseteq f(w')$, as desired. □

Our semantics uses a general method for lifting a complete Heyting algebra structure and a monotone commutative monoid of *UAdm* to those of functions to *UAdm*. To describe this construction, we let $(A, \circ, e)$ be a monoid such that A is an object in *CBUlt* and $\circ$ is a non-expansive operation, and with the induced preorder $a \sqsubseteq a' \;\Leftrightarrow\; \exists a_0 \in A.\ a \circ a_0 = a'$. Also, let $(B, \sqsubseteq)$ be a complete Heyting algebra $(B, \sqsubseteq)$ such that B is an object in *CBUlt* and all the algebra operations are non-expansive. Finally, assume a monotone commutative monoid $(B, \mathsf{e}, *)$ such that the $*$ operator is non-expansive. The method is described by the lemma below.

Lemma 17 (Lifting of algebra structure). *If the set $A \to_m B$ of monotone non-expansive functions with the* sup *metric satisfies the completeness condition from metric spaces, then it forms a complete Heyting algebra with a monotone commutative monoid, where the operations are non-expansive. Futhermore, in that case, all the algebra operators of $A \to_m B$ except for $\Rightarrow$ are given by the pointwise extension of the corresponding operators in B, and the remaining $\Rightarrow$ is defined as follows:*

$$(f \Rightarrow g)(a) = \bigsqcap_{a'} (f(a \circ a') \Rightarrow g(a \circ a')).$$

Proof. Since the set of non-expansive functions from A to B form a complete 1-bounded ultrametric space, the completeness assumption in the lemma implies that $A \to_m B$ is also a complete 1-bounded ultrametric space. Thus, it is sufficient to prove that $A \to_m B$ is a complete Heyting algebra with a monotone commutative monoid where its carrier is an object in *CBUlt* and all the operations are non-expansive.

We start by showing that all the claimed algebra operators of $A \to_m B$ by this lemma are well-defined. The easiest cases are the various units

$$\mathsf{e}(a) \;=\; \mathsf{e}, \qquad \bot(a) \;=\; \bot, \qquad \top(a) \;=\; \top,$$

where the constants on the RHS of the equations are the units in B. All these units of $A \to_m B$ are constant functions, so they are monotone and non-expansive.

Next, let $\oplus$ be one of $\{*, \sqcup, \sqcap\}$ and let $\bigoplus$ be in $\{\bigsqcup, \bigsqcap\}$. Then, the lemma says that

$$(f \oplus g)(a) \;=\; f(a) \oplus g(a) \quad \text{and} \quad (\bigoplus_{i \in I} f_i)(a) \;=\; (\bigoplus_{i \in I} f(a)),$$

where f, g and the f_i are monotone non-expansive functions from A to B and the operators on the RHS of the equations are from the assumed algebraic structure of B. Furthermore, the axioms of a Heyting algebra and a monotone commutative monoid imply that the operators $\oplus$ and $\bigoplus$ of B are monotone. The desired well-definedness follows from the monotonicity and non-expansiveness of $\oplus$ and $\bigoplus$ of B. We will show the $\oplus$ case only, since that for $\bigoplus$ is similar. When $a \sqsubseteq b$, we have that

$$(f \oplus g)(a) \;=\; (f(a) \oplus g(a)) \;\sqsubseteq\; (f(b) \oplus g(b)) \;=\; (f \oplus g)(b),$$

where the $\sqsubseteq$ relationship in the middle comes from the monotonicity of $\oplus$ of B. This shows the monotonicity of $f \oplus g$. Also, for all $a, b \in A$,

$$\begin{aligned} d((f \oplus g)(a),\, (f \oplus g)(b)) &= d(f(a) \oplus g(a),\, f(b) \oplus g(b)) \\ &\leq \max\{d(f(a), f(b)),\, d(g(a), g(b))\} \\ &\leq \max\{d(a, b),\, d(a, b)\} \\ &= d(a, b). \end{aligned}$$

The $\leq$ in the second line is by the non-expansiveness of $\oplus$ of B and the $\leq$ in the third line is by the non-expansiveness of f and g. Thus, $f \oplus g$ is non-expansive. Finally, for all f, f', g, g' in $A \rightarrow_m B$ and all a,

$$\begin{aligned} d((f \oplus g)(a),\, (f' \oplus g')(a)) &= d(f(a) \oplus g(a),\, f'(a) \oplus g'(a)) \\ &\leq \max\{d(f(a), f'(a)),\, d(g(a), g'(a))\} \\ &\leq \max\{d(f, f'),\, d(g, g')\} \\ &= d((f, g),\, (f', g')). \end{aligned}$$

Thus, the $\oplus$ operator of $A \rightarrow_m B$ itself is non-expansive.

We now move on to the well-definedness of the other operator $\Rightarrow$. Consider a, b in A such that $a \sqsubseteq b$, which means that $b = a \circ a_0$ for some a_0 in A.

$$\begin{aligned} (f \Rightarrow g)(a) &= \textstyle\bigsqcap_{a'}(f(a \circ a') \Rightarrow g(a \circ a')) \\ &\sqsubseteq \textstyle\bigsqcap_{a'}(f(a \circ (a_0 \circ a')) \Rightarrow g(a \circ (a_0 \circ a')) \\ &= \textstyle\bigsqcap_{a'}(f((a \circ a_0) \circ a') \Rightarrow g((a \circ a_0) \circ a')) \\ &= \textstyle\bigsqcap_{a'}(f(b \circ a') \Rightarrow g(b \circ a')) \\ &= (f \Rightarrow g)(b). \end{aligned}$$

Thus, $f \Rightarrow g$ is monotone. Also, for all a_1, a_2 in A, we have that

$$\begin{aligned} & d((f \Rightarrow g)(a_1),\, (f \Rightarrow g)(a_2)) \\ &= d(\textstyle\bigsqcap_{a'}(f(a_1 \circ a') \Rightarrow g(a_1 \circ a')),\, \bigsqcap_{a'}(f(a_2 \circ a') \Rightarrow g(a_2 \circ a'))) \\ &\leq \max\{d(f(a_1 \circ a') \Rightarrow g(a_1 \circ a'),\, f(a_2 \circ a') \Rightarrow g(a_2 \circ a')) \mid a' \in A\} \\ &\leq \max\{d(f(a_1 \circ a'),\, f(a_2 \circ a')),\quad d(g(a_1 \circ a'),\, g(a_2 \circ a')) \mid a' \in A\} \\ &\leq \max\{d(a_1 \circ a',\, a_2 \circ a') \mid a' \in A\} \\ &\leq d(a_1,\, a_2). \end{aligned}$$

The first two $\leq$'s are by the non-expansiveness of $\bigsqcap$ and $\Rightarrow$ of A, the third follows from the non-expansiveness of f and g, and the last holds because of the non-expansiveness of $\circ$. Note that we have just shown the non-expansiveness of $f \Rightarrow g$. Finally, the non-expansiveness of the $\Rightarrow$ operator itself can be proved as

follows: for all $f, g, f', g' \in A \to_m B$ and all $a \in A$,

$$\begin{aligned}
& d((f \Rightarrow g)(a),\ (f' \Rightarrow g')(a)) \\
=\ & d(\textstyle\bigsqcap_{a'}(f(a \circ a') \Rightarrow g(a \circ a')),\ \bigsqcap_{a'}(f'(a \circ a') \Rightarrow g'(a \circ a'))) \\
\leq\ & \max\{d(f(a \circ a') \Rightarrow g(a \circ a'),\ f'(a \circ a') \Rightarrow g'(a \circ a'))\ \mid\ a' \in A\} \\
\leq\ & \max\{d(f(a \circ a'),\ f'(a \circ a')),\quad d(g(a \circ a'),\ g'(a \circ a'))\ \mid\ a' \in A\} \\
\leq\ & \max\{d(f, f'),\quad d(g, g')\ \mid\ a' \in A\} \\
\leq\ & d((f, g),\ (f', g')).
\end{aligned}$$

Now, it remains to show that these defined operators in $A \to_m B$ satisfy all the axioms of a complete Heyting algebra with a monotone commutative monoid. Showing this is easy, except for the axioms for the operator $\Rightarrow$. Since all the other operators are defined pointwise, one only needs to unroll their definitions and use the fact that B is a complete Heyting algebra with a monotone commutative monoid. Thus, we focus on showing the axiom for $\Rightarrow$: for all f, g, k in $A \to_m B$,

$$(f \wedge g) \sqsubseteq k \iff f \sqsubseteq (g \Rightarrow k).$$

To show the only-if direction, suppose that $(f \wedge g) \sqsubseteq k$ and pick $a \in A$. By the definition of $g \Rightarrow k$, it is sufficient to show that

$$\forall a_0 \in A.\quad f(a)\ \sqsubseteq\ g(a \circ a_0) \Rightarrow k(a \circ a_0). \tag{10}$$

Since f is monotone and $a \sqsubseteq a \circ a_0$, we have that

$$f(a)\ \sqsubseteq\ f(a \circ a_0). \tag{11}$$

Furthermore, from the assumption that $(f \wedge g) \sqsubseteq k$, it follows that $f(a \circ a_0) \wedge g(a \circ a_0) \sqsubseteq k(a \circ a_0)$, and this followed relationship implies

$$f(a \circ a_0)\ \sqsubseteq\ g(a \circ a_0) \Rightarrow k(a \circ a_0). \tag{12}$$

The required (10) follows from (11) and (12). For the if direction, assume that $f \sqsubseteq (g \Rightarrow k)$ and pick $a \in A$. Then, by this assumption,

$$f(a)\quad \sqsubseteq\quad (\bigsqcap_{a'} g(a \circ a') \Rightarrow k(a \circ a'))\quad \sqsubseteq\quad g(a \circ e) \Rightarrow k(a \circ e)\quad =\quad g(a) \Rightarrow k(a).$$

Note that this order relationship is in B, where $\Rightarrow$ satisfies the required axiom. Thus, the above implies that

$$(f \wedge g)(a)\quad =\quad f(a) \wedge g(a)\quad \sqsubseteq\quad k(a).$$

Since a is chosen arbitrarily, this means that $f \wedge g \sqsubseteq k$, as desired. □

C Recursive and monotone worlds in Section 4

In this appendix, we give the details of Theorem 11, i.e., we construct an object W in *CBUlt* that satisfies the equation

$$\iota : W \;\cong\; (\tfrac{1}{2} \cdot W) \to_m UAdm \;=\; Assert. \tag{13}$$

The order on (13) is induced by the following monoid structure on W:

$$w_1 \circ w_2 = \iota^{-1}\big(\lambda w.(\iota(w_1)(w_2 \circ w) \;*\; \iota(w_2)(w))\big) \qquad e = \iota^{-1}(\lambda w.\mathsf{e}).$$

Note that $\circ$ satisfies a recursive equation that makes use of the isomorphism ι^{-1} between $Assert$ and W, and that the monotonicity requirements on $Assert$ and W stated by (13) already use this operation $\circ$.

Conceptually, it is useful to rewrite $\circ$ in terms of the following operation $\otimes : Assert \times W \to Assert$ (in the first line, $*$ is the separating conjunction lifted from $UAdm$ to $Assert$):

$$w_1 \circ w_2 = \iota^{-1}\big(\iota(w_1) \otimes w_2 \;*\; \iota(w_2))\big)$$
$$p \otimes w = \lambda w'.\, p(w \circ w')$$

It turns out that this operation is a (right-) action of the monoid W on $Assert$, i.e., $(p \otimes w_1) \otimes w_2 = p \otimes (w_1 \circ w_2)$ and $p \otimes e = p$ for all $p \in Assert$ and $w_1, w_2 \in W$. The first of these equations is by the associativity of $\circ$, and the second by the fact that e is a right unit for $\circ$.

We will use this conceptual view to organize the following explicit construction of $\iota^{-1} : Assert \cong W$, but must weaken it for our proofs: instead of working with monoids X in *CBUlt*, we only require that associativity holds up to some approximation 2^{-k}, and similarly we do not require an isomorphism between X and $\frac{1}{2} \cdot X \to_m UAdm$ but only that an inverse up to approximation 2^{-k} is given.

Preliminaries. Recall that the heap model used in Section 4 satisfies a number of properties that go beyond the general requirements of the framework described in Section 2. First, there is an empty heap $\{\!|\}\!\}$, which provides a unit for the combination operation: $h \cdot \{\!|\}\!\} = \{\!|\}\!\} \cdot h = h$. Next, $\bot$ is a zero for the combination operation: $h \cdot \bot = \bot \cdot h = \bot$. Using the empty heap we can give a unit for the separating conjunction, $\mathsf{e} \stackrel{def}{=} \{\bot, \{\!|\}\!\}\}$, in $UAdm$. This set satisfies $\pi_k(\mathsf{e}) = \mathsf{e}$ for all $k > 0$.

In the following proofs, the notation $f \cdot g$ is used for function composition, to avoid confusion with the monoid operation $\circ$. Moreover we use the notation $x \stackrel{n}{=} y$ to mean $d(x, y) \leq 2^{-n}$. Note that, because of the ultrametric triangle inequality, each $\stackrel{n}{=}$ is an equivalence relation.

C.1 Cauchy tower

We define an ω^{op}-chain of metric spaces $(W_k, \iota_k^\circ)_k$ in *CBUlt*,

$$W_0 \underset{\iota_0^\circ}{\overset{\iota_0}{\rightleftarrows}} W_1 \underset{\iota_1^\circ}{\overset{\iota_1}{\rightleftarrows}} W_2 \underset{\iota_2^\circ}{\overset{\iota_2}{\rightleftarrows}} \cdots \underset{\iota_k^\circ}{\overset{\iota_k}{\rightleftarrows}} W_{k+1} \underset{\iota_{k+1}^\circ}{\overset{\iota_{k+1}}{\rightleftarrows}} \cdots \tag{14}$$

each equipped with a preorder $\sqsubseteq_k$ and with (non-expansive) operations

$$\circ_k : W_k \times W_k \to W_k, \qquad \otimes_k : Assert_k \times W_k \to Assert_k, \qquad e_k \in W_k$$

for $Assert_k = \frac{1}{2} \cdot W_k \to_m UAdm$. These data will satisfy the following properties:

C0 $W_{k+1} = Assert_k$ and $x \circ_{k+1} y = x \otimes_k \iota_k^\circ(y) * y$ and $e_{k+1} = \lambda w.\mathsf{e}$.
C1 ι_k° and ι_k are non-expansive maps.
C2 $d(\iota_k^\circ \cdot \iota_k, id_{W_k}) \leq 2^{-k}$ and $d(\iota_k \cdot \iota_k^\circ, id_{Assert_k}) \leq 2^{-k}$.
C3 $(x \circ_k y) \circ_k z \overset{k}{=} x \circ_k (y \circ_k z)$.
C4 e_k is the left and right unit for $\circ_k$, such that $\iota_k(e_k) = \lambda x.\mathsf{e}$.
C5 $(p \otimes_k x)(y) = p(x \circ_k y)$.
C6 $\iota_k^\circ(x \otimes_k \iota_k^\circ(y) * y) \overset{k}{=} (\iota_k^\circ x) \circ_k (\iota_k^\circ y)$ and $\iota_k(x \circ_k y) \overset{k}{=} (\iota_k x) \otimes_k (\iota_k^\circ(\iota_k(y))) * (\iota_k y)$ and $\iota_k^\circ(\lambda w.\mathsf{e}) = e_k$.

Note that the third property means that (14) is a Cauchy tower. Also note that C0 gives the definition of $(W_{k+1}, e_{k+1}, \circ_{k+1})$, and that we can *define* $\otimes_k$ from $\circ_k$ according to property C5. In the following, we will therefore omit the explicit verification of these two properties. Finally, after defining W_{k+1} as $Assert_k$ and defining $\circ_{k+1}$ using C0, the three statements in C6 can be rewritten more naturally as

$$\iota_k^\circ(p \circ_{k+1} p') \overset{k}{=} (\iota_k^\circ p) \circ_k (\iota_k^\circ p'), \qquad \iota_k(w \circ_k w') \overset{k}{=} (\iota_k w) \circ_{k+1} (\iota_k w'),$$
$$\iota_k^\circ(e_{k+1}) = e_k.$$

The case $k = 0$. We define $W_0 \overset{def}{=} \mathbf{1}$ to be the one-point space, with the evident operation $\langle\rangle \circ_0 \langle\rangle = \langle\rangle$, with $e_0 = \langle\rangle$, and with the trivial order $\langle\rangle \sqsubseteq_0 \langle\rangle$. Note that the requirement $(p \otimes_0 x)(y) = p(x \circ_0 y)$ here simply amounts to the trivial action $p \otimes_0 x = p$ on $p : \frac{1}{2} \cdot \mathbf{1} \to_m UAdm$. Finally, we define two maps ι_0° and ι_0 by

$$\iota_0^\circ : (\tfrac{1}{2} \cdot \mathbf{1} \to_m UAdm) \to \mathbf{1} \qquad \iota_0 : \mathbf{1} \to (\tfrac{1}{2} \cdot \mathbf{1} \to_m UAdm)$$
$$\iota_0^\circ(p) = \langle\rangle \qquad \iota_0(w) = \lambda u.\,\mathsf{e}$$

Note that ι_0° and ι_0 are non-expansive and monotonic maps of the right type, and that $\iota_0^\circ \cdot \iota_0 = id_{\mathbf{1}}$ and $(\iota_0 \cdot \iota_0^\circ)(p) \overset{0}{=} p$ for all $p \in \frac{1}{2} \cdot \mathbf{1} \to_m UAdm$, by definition of the sup-metric and since all distances in $UAdm$ are bounded by 1.

The case $k > 0$. For all k we now define $W_{k+1} \overset{def}{=} \frac{1}{2} \cdot W_k \to_m UAdm$, where $\to_m$ denotes the set of non-expansive and monotone functions equipped with the sup-metric (cf. Lemma 16). On W_{k+1} the operation $\circ_{k+1}$ and the unit e_{k+1} are defined according to C0:

$$p \circ_{k+1} q = p \otimes_k (\iota_k^\circ q) * q, \qquad e_{k+1} = \iota_k(e_k).$$

The preorder $\sqsubseteq_{k+1}$ is defined by

$$p \sqsubseteq_{k+1} q \;\Leftrightarrow\; \exists p_0.\; q \overset{k+1}{=} p \circ_{k+1} p_0.$$

Finally, the maps ι°_{k+1} and ι_{k+1} are given by:

$$\iota^\circ_{k+1} : (\tfrac{1}{2} \cdot W_{k+1} \to_m \mathit{UAdm}) \to W_{k+1} \qquad \iota_{k+1} : W_{k+1} \to (\tfrac{1}{2} \cdot W_{k+1} \to_m \mathit{UAdm})$$
$$\iota^\circ_{k+1}(p) = \pi_{k+1} \cdot p \cdot \iota_k \qquad \iota_{k+1}(w) = \pi_{k+2} \cdot w \cdot \iota^\circ_k$$

Verification of properties C1–C6. From what we have said above, it is clear that all the conditions C1–C6 hold in the case $k = 0$. In the remainder of this subsection we now show that C1–C6 also hold for $k > 0$, under the assumption that they hold for $k - 1$. In particular our proofs will show that ι°_k, ι_k and $\sqsubseteq_k$ are well-defined.

Lemma 18 (Non-expansiveness of ι°_k and ι_k). *Let $k > 0$. Then, ι°_k and ι_k are non-expansive maps between $\frac{1}{2} \cdot W_k \to \mathit{UAdm}$ and W_k.*

Proof. Recall that ι°_k and ι_k are given by composing with the non-expansive maps π_k, π_{k+1}, ι_{k-1} and ι°_{k-1}. From this, it follows that the images of these functions consist of non-expansive functions only. The non-expansiveness of the operations ι°_k and ι_k themselves holds as well, because in addition to π_k and π_{k+1}, the function composition is non-expansive. □

To conclude that property C1 holds, we also need to show that ι_k and ι°_k map into the subsets of monotonic maps. The proof of this property is deferred to Lemma 23 because we first establish that $\sqsubseteq_k$ indeed is a preorder on W_k.

Lemma 19 (Non-expansiveness of $\circ_k$). *For $k > 0$ and all $p, q \in W_k$, $p \circ_k q$ is an element of W_k. Moreover, $\circ_k$ is a non-expansive operation on W_k.*

Proof. Let $p, q \in W_k$. We must show that $p \circ_k q \in W_k = \frac{1}{2} \cdot W_{k-1} \to_m \mathit{UAdm}$. By definition,

$$(p \circ_k q)(w) \;=\; p((\iota^\circ_{k-1}\, q) \circ_{k-1} w) * q(w).$$

First, $p \circ_k q$ is a non-expansive map from $\frac{1}{2} \cdot W_{k-1} \to \mathit{UAdm}$: If $w \overset{n}{=} w'$ in $\frac{1}{2} \cdot W_{k-1}$ then $(\iota^\circ_{k-1}\, q) \circ_{k-1} w \overset{n}{=} (\iota^\circ_{k-1}\, q) \circ_{k-1} w'$ in $\frac{1}{2} \cdot W_{k-1}$ and also $q(w) \overset{n}{=} q(w')$, since by assumption $\circ_{k-1}$ and q are non-expansive. Therefore, by the non-expansiveness of p and $*$, we obtain

$$(p \circ_k q)(w) \overset{n}{=} (p \circ_k q)(w').$$

Next, $p \circ_k q$ is monotone: Suppose $w \sqsubseteq_{k-1} w'$, so there exists $w_0 \in \frac{1}{2} \cdot W_{k-1}$ such that $w' \overset{k-1}{=} w \circ_{k-1} w_0$ in W_{k-1}. We must show that $(p \circ_k q)(w) \subseteq (p \circ_k q)(w')$. By

assumption, q is monotone, so $q(w) \subseteq q(w')$. Moreover, by the non-expansiveness of $\circ_{k-1}$ and the approximate associativity (C3) we have

$$\begin{aligned}(\iota^\circ_{k-1}\, q) \circ_{k-1} w' &\overset{k-1}{=} (\iota^\circ_{k-1}\, q) \circ_{k-1} (w \circ_{k-1} w_0)\\ &\overset{k-1}{=} ((\iota^\circ_{k-1}\, q) \circ_{k-1} w) \circ_{k-1} w_0\\ &\sqsupseteq_{k-1} (\iota^\circ_{k-1}\, q) \circ_{k-1} w,\end{aligned}$$

and hence $p((\iota^\circ_{k-1}\, q) \circ_{k-1} w) \subseteq p((\iota^\circ_{k-1}\, q) \circ_{k-1} w')$ by the assumption that p is monotone. The monotonicity of $p \circ_k q$ therefore follows from the monotonicity of $*$ on $UAdm$ (Lemma 15).

Finally, $\circ_k$ is non-expansive operation on W_k: Suppose $p \overset{n}{=} p'$ and $q \overset{n}{=} q'$ in W_k. By the non-expansiveness of composition and the non-expansiveness of $\circ_{k-1}$ and ι°_{k-1},

$$(\iota^\circ_{k-1}\, q) \circ_{k-1} w \overset{n}{=} (\iota^\circ_{k-1}\, q') \circ_{k-1} w$$

holds for all $w \in W_{k-1}$. This means that in $\frac{1}{2} \cdot W_{k-1}$, the below holds:

$$\forall w \in \tfrac{1}{2} \cdot W_{k-1}.\quad (\iota^\circ_{k-1}\, q) \circ_{k-1} w \overset{n+1}{=} (\iota^\circ_{k-1}\, q') \circ_{k-1} w.$$

By the non-expansiveness of $*, p, p'$, and the assumption that $p \overset{n}{=} p'$ and $q \overset{n}{=} q'$, this shows that

$$\begin{aligned}(p \circ_k q)(w) &= p((\iota^\circ_{k-1}\, q) \circ_{k-1} w) * q(w)\\ &\overset{n}{=} p'((\iota^\circ_{k-1}\, q') \circ_{k-1} w) * q'(w) = (p' \circ_k q')(w).\end{aligned}$$

Since this holds for arbitrary w, the sup metric on $\frac{1}{2} \cdot W_{k-1} \to_m UAdm$ yields $p \circ_k q \overset{n}{=} p' \circ_k q'$. □

Lemma 20 (Associativity). *Suppose $k > 0$. Then $(p \circ_k p') \circ_k p'' \overset{k}{=} p \circ_k (p' \circ_k p'')$ for all $p, p', p'' \in W_k$.*

Proof. Let $w \in \frac{1}{2} \cdot W_{k-1}$. We have

$$\begin{aligned}&((p \circ_k p') \circ_k p'')(w)\\ &\quad= (p \circ_k p')((\iota^\circ_{k-1}\, p'') \circ_{k-1} w) * p''(w)\\ &\quad= p((\iota^\circ_{k-1}\, p') \circ_{k-1} ((\iota^\circ_{k-1}\, p'') \circ_{k-1} w)) * p'((\iota^\circ_{k-1}\, p'') \circ_k w) * p''(w)\\ &\quad= p((\iota^\circ_{k-1}\, p') \circ_{k-1} ((\iota^\circ_{k-1}\, p'') \circ_{k-1} w)) * (p' \circ_k p'')(w)\\ &\quad\overset{k}{=} p(((\iota^\circ_{k-1}\, p') \circ_{k-1} (\iota^\circ_{k-1}\, p'')) \circ_{k-1} w) * (p' \circ_k p'')(w)\\ &\quad\overset{k}{=} p(\iota^\circ_{k-1}(p' \circ_k p'') \circ_{k-1} w) * (p' \circ_k p'')(w)\\ &\quad= (p \circ_k (p' \circ_k p''))(w).\end{aligned}$$

Here, the first $\overset{k}{=}$ equality is by assumption C3, the scaling by $1/2$ and the non-expansiveness of p, and the following $\overset{k}{=}$ equality is by assumption C6 and the scalining by $1/2$ and the non-expansiveness of p. Since w was chosen arbitrarily, the sup metric on W_k therefore gives $(p \circ_k p') \circ_k p'' \overset{k}{=} p \circ_k (p' \circ_k p'')$. □

Thus we have established property C3. Property C4 is the following lemma:

Lemma 21 (Unit). *Suppose $k > 0$. Let e_k be defined by $e_k(w) = \mathsf{e}$ for all $w \in \frac{1}{2} \cdot W_{k-1}$. Then $\iota_k(e_k) = \lambda x.\mathsf{e}$, and $p \circ_k e_k = p$ and $e_k \circ_k q = q$ holds for all $p, q \in W_k$.*

Proof. Since $k > 0$, $\pi_k(\mathsf{e}) = \pi_{k+1}(\mathsf{e}) = \mathsf{e}$. Thus, it is easy to see that $\iota_k(e_k) = \pi_{k+1} \cdot e_k \cdot \iota^\circ_{k-1} = \lambda x.\mathsf{e}$. Next, let $w \in \frac{1}{2} \cdot W_{k-1}$. Then by definition of e_k and $\circ_k$,

$$(e_k \circ_k p)(w) = e_k((\iota^\circ_{k-1}\, p) \circ_{k-1} w) * p(w) = \mathsf{e} * p(w) = p(w),$$

i.e., e_k is a left unit for $\circ_k$. It remains to show that e_k is also a right unit. For this, note that when $k = 1$, $\iota^\circ_{k-1}(e_k) = \langle\rangle = e_{k-1}$, and that when $k - 1 > 0$, $\iota^\circ_{k-1}(e_k)(w) = (\pi_{k-1} \cdot e_k \cdot \iota_{k-2})(w) = \mathsf{e} = e_{k-1}(w)$. Thus, in both cases, we have that $\iota^\circ_{k-1}(e_k) = e_{k-1}$. This implies that

$$(p \circ_k e_k)(w) = p((\iota^\circ_{k-1}\, e_k) \circ_{k-1} w) * e_k(w) = p(e_{k-1} \circ_{k-1} w) * \mathsf{e} = p(w)$$

by the definition of $\circ_k$, and by $e_{k-1} \circ_{k-1} w = w$. □

Lemma 22 (Preorder). *The relation $\sqsubseteq_k$ is a preorder.*

Proof. A consequence of Lemmas 21 and 20. □

We now show that property C1 holds:

Lemma 23 (Monotonicity). *Suppose $k > 0$. For all $p \in W_k$ and $q \in W_{k+1}$, $\iota_k(p)$ and $\iota^\circ_k(q)$ are monotonic maps. That is, ι_k maps into the complete subspace $Assert_k = \frac{1}{2} \cdot W_k \to_m UAdm$ of monotonic maps, and ι°_k into the subspace $W_k = Assert_{k-1} = \frac{1}{2} \cdot W_{k-1} \to_m UAdm$ of monotonic maps.*

Proof. Let $p \in W_k$, and let $w_1, w_2 \in \frac{1}{2} \cdot W_k$ such that $w_1 \sqsubseteq_k w_2$. We first show that $\iota_k(p)(w_1) \subseteq \iota_k(p)(w_2)$. By definition, there exists some $w_0 \in W_k$ such that $w_2 \overset{k}{=} w_1 \circ_k w_0$. By assumption C6 and the non-expansiveness of ι°_{k-1},

$$\iota^\circ_{k-1}(w_2) \overset{k}{=} \iota^\circ_{k-1}(w_1 \circ_k w_0) \overset{k-1}{=} \iota^\circ_{k-1}(w_1) \circ_{k-1} \iota^\circ_{k-1}(w_0)$$

and hence $\iota^\circ_{k-1}(w_1) \sqsubseteq_{k-1} \iota^\circ_{k-1}(w_2)$. By the monotonicity of p we thus obtain

$$p(\iota^\circ_{k-1}(w_1)) \subseteq p(\iota^\circ_{k-1}(w_2)).$$

Since π_{k+1} preserves the subset order, the subset relationship above implies that

$$\iota_k(p)(w_1) = (\pi_{k+1} \cdot p \cdot \iota^\circ_{k-1})(w_1) \subseteq (\pi_{k+1} \cdot p \cdot \iota^\circ_{k-1})(w_2) = \iota_k(p)(w_2)$$

as required.

Next, we move on to the claimed property of ι°_k. Let $q \in W_{k+1}$, and let $w_1, w_2 \in \frac{1}{2} \cdot W_{k-1}$ such that $w_1 \sqsubseteq_{k-1} w_2$. We must prove that $\iota^\circ_k(q)(w_1) \subseteq$

$\iota_k^\circ(q)(w_2)$. By definition, there exists some $w_0 \in W_{k-1}$ such that $w_2 \overset{k-1}{=} w_1 \circ_{k-1} w_0$. By the non-expansiveness of ι_{k-1} and assumption C6,

$$\iota_{k-1}(w_2) \overset{k-1}{=} \iota_{k-1}(w_1 \circ_{k-1} w_0) \overset{k-1}{=} \iota_{k-1}(w_1) \circ_k \iota_{k-1}(w_0).$$

By the contractiveness of q, we thus obtain

$$q(\iota_{k-1}(w_2)) \overset{k}{=} q(\iota_{k-1}(w_1) \circ_k \iota_{k-1}(w_0)).$$

This implies the (exact) equality below, because π_k maps $\overset{k}{=}$-equivalent elements to the same value:

$$\iota_k^\circ(q)(w_2) \;=\; \pi_k(q(\iota_{k-1}(w_2))) \;=\; \pi_k(q(\iota_{k-1}(w_1) \circ_k \iota_{k-1}(w_0))).$$

Now, it remains to prove that the right most set in the above equalities includes $\iota_k^\circ(q)(w_1)$. We discharge this remaining proof obligation as follows. Since q is monotone, we have that

$$q(\iota_{k-1}(w_1) \circ_k \iota_{k-1}(w_0)) \;\supseteq\; q(\iota_{k-1}(w_1)).$$

Then, since π_k preserves the subset order, we have the required subset relationship below:

$$\pi_k(q(\iota_{k-1}(w_1) \circ_k \iota_{k-1}(w_0))) \;\supseteq\; \pi_k(q(\iota_{k-1}(w_1))) \;=\; \iota_k^\circ(q)(w_1).$$

□

Lemma 24 (Cauchy tower). *Let $k > 0$. We have:*

1. $d(\iota_k^\circ \cdot \iota_k,\, id_{W_k}) \leq 2^{-k}$.
2. $d(\iota_k \cdot \iota_k^\circ,\, id_{Assert_k}) \leq 2^{-k}$

Proof. For the first part, let $w \in W_k$. By definition,

$$(\iota_k^\circ \cdot \iota_k)(w) \;=\; \pi_k \cdot (\pi_{k+1} \cdot w \cdot \iota_{k-1}^\circ) \cdot \iota_{k-1} \;=\; (\pi_k \cdot \pi_{k+1}) \cdot w \cdot (\iota_{k-1}^\circ \cdot \iota_{k-1}).$$

Recall that w is a non-expansive function from $\frac{1}{2} \cdot W_{k-1}$ to $UAdm$. That is, it is a contractive map (at least by $1/2$) on the unscaled domain W_{k-1}. Furthermore, $d(\iota_{k-1}^\circ \cdot \iota_{k-1}, id_{W_{k-1}}) \leq 2^{-(k-1)}$ by assumption C2. Thus, we have that

$$d(w \cdot (\iota_{k-1}^\circ \cdot \iota_{k-1}),\; w \cdot id_{W_{k-1}}) \;\leq\; 2^{-k}.$$

From this, $d(\pi_k \cdot \pi_{k+1}, id_{UAdm}) \leq 2^{-k}$ and the non-expansiveness of the function composition, it follows that

$$d((\pi_k \cdot \pi_{k+1}) \cdot w \cdot (\iota_{k-1}^\circ \cdot \iota_{k-1}),\;\; id_{UAdm} \cdot w \cdot id_{W_{k-1}}) \;\leq\; 2^{-k}.$$

Thus, we have that $d((\iota_k^\circ \cdot \iota_k)(w),\, w) \leq 2^{-k}$. Since w was chosen arbitrarily, $d(\iota_k^\circ \cdot \iota_k, id_{W_k}) \leq 2^{-k}$ follows.

For the second part, let $p \in Assert_k = \frac{1}{2} \cdot W_k \to_m UAdm$. By the assumption C2,

$$d(\iota_{k-1} \cdot \iota^\circ_{k-1},\ id_{Assert_{k-1}}) \ \leq \ 2^{-(k-1)},$$

and p is contractive at least by $1/2$ with respect to the original metric in W_k. Thus,

$$d(p \cdot (\iota_{k-1} \cdot \iota^\circ_{k-1}),\ p \cdot id_{Assert_{k-1}}) \ \leq \ 2^{-k},$$

Furthermore, $d(\pi_{k+1} \cdot \pi_k,\ id_{UAdm}) \leq 2^{-k}$, and by definition

$$(\iota_k \cdot \iota^\circ_k)(p) \ = \ \pi_{k+1} \cdot (\pi_k \cdot p \cdot \iota_{k-1}) \cdot \iota^\circ_{k-1} \ = \ (\pi_{k+1} \cdot \pi_k) \cdot p \cdot (\iota_{k-1} \cdot \iota^\circ_{k-1}).$$

Thus, the non-expansiveness of function composition implies that

$$d((\iota_k \cdot \iota^\circ_k)(p),\ id_{UAdm} \cdot p \cdot id_{Assert_{k-1}}) \ \leq \ 2^{-k}.$$

Since p is chosen arbitrary, we can conclude that $d(\iota_k \cdot \iota^\circ_k, id_{W_{k+1}}) \leq 2^{-k}$ by the sup metric on $Assert_k \to Assert_k$. □

This proves C2. Finally, we establish property C6:

Lemma 25 (Tensor preservation). *Suppose $k > 0$.*

1. *For all $p, p' \in Assert_k$: $\iota^\circ_k(p \otimes_k (\iota^\circ_k\, p') \ast p') \overset{k}{=} (\iota^\circ_k\, p) \circ_k (\iota^\circ_k\, p')$.*
2. *For all $w, w' \in W_k$: $\iota_k(w \circ_k w') \overset{k}{=} (\iota_k\, w) \otimes_k (\iota^\circ_k(\iota_k(w'))) \ast (\iota_k\, w')$.*
3. *$\iota^\circ_k(\lambda w.\, \mathsf{e}) = e_k$.*

Note that, after defining W_{k+1} as $Assert_k$ and after defining $\circ_{k+1}$, the two statements can be rewritten more naturally as $\iota^\circ_k(p \circ_{k+1} p') \overset{k}{=} (\iota^\circ_k\, p) \circ_k (\iota^\circ_k\, p')$ and $\iota_k(w \circ_k w') \overset{k}{=} (\iota_k\, w) \circ_{k+1} (\iota_k\, w')$.

Proof. For the first statement, let $w \in W_{k-1}$. By definition of $\circ_k$ we have

$$\begin{aligned}
((\iota^\circ_k\, p) \circ_k (\iota^\circ_k\, p'))(w) \ &= \ ((\iota^\circ_k\, p) \otimes_{k-1} (\iota^\circ_{k-1}(\iota^\circ_k\, p')))(w) \ast (\iota^\circ_k\, p')(w) \\
&= \ (\iota^\circ_k\, p)((\iota^\circ_{k-1}(\iota^\circ_k\, p')) \circ_{k-1} w) \ast (\iota^\circ_k\, p')(w) \\
&= \ (\pi_k \cdot p \cdot \iota_{k-1})((\iota^\circ_{k-1}(\iota^\circ_k\, p')) \circ_{k-1} w) \ast (\pi_k \cdot p' \cdot \iota_{k-1})(w) \\
&= \ \pi_k\Big(p\big(\iota_{k-1}\big((\iota^\circ_{k-1}(\iota^\circ_k\, p')) \ \circ_{k-1} \ w\big)\big) \ \ast \ p'(\iota_{k-1} w)\Big).
\end{aligned}$$

By assumption C6, ι_{k-1} distributes over $\circ_{k-1}$, so that by scaling we obtain the following approximate equality in $\frac{1}{2} \cdot W_k$:

$$\iota_{k-1}((\iota^\circ_{k-1}(\iota^\circ_k\, p')) \circ_{k-1} w) \ \overset{k}{=} \ (\iota_{k-1}(\iota^\circ_{k-1}(\iota^\circ_k\, p'))) \otimes_{k-1} (\iota^\circ_{k-1}(\iota_{k-1} w)) \ \ast \ (\iota_{k-1}\, w).$$

Moreover, we have that $d(\iota_{k-1} \cdot \iota^\circ_{k-1}, id_{W_k}) \leq 2^{-(k-1)}$ by property C2, which means $d(\iota_{k-1} \cdot \iota^\circ_{k-1}, id_{1/2 \cdot W_k}) \leq 2^{-k}$. Hence, by the non-expansiveness of $\otimes_{k-1}$,

of p, and of the lifted $*$, we can continue thus:

$$\begin{aligned}
&\pi_k\Big(p\big(\iota_{k-1}\big((\iota^\circ_{k-1}(\iota^\circ_k p'))\ \circ_{k-1}\ w\big)\big) * p'(\iota_{k-1}\, w)\Big)\\
&\quad= \pi_k\Big(p\big(\iota_{k-1}\big((\iota^\circ_{k-1}(\iota^\circ_k p'))\ \circ_{k-1}\ w\big)\big)\Big) * \pi_k\Big(p'(\iota_{k-1}\, w)\Big)\\
&\quad\overset{k}{=} \pi_k\Big(p((\iota_{k-1}(\iota^\circ_{k-1}(\iota^\circ_k p'))) \otimes_{k-1} (\iota^\circ_{k-1}(\iota_{k-1}(w)))\ *\ (\iota_{k-1}\, w))\Big) * \pi_k\Big(p'(\iota_{k-1}\, w)\Big)\\
&\quad\overset{k}{=} \pi_k\Big(p((\iota^\circ_k p') \otimes_{k-1} (\iota^\circ_{k-1}(\iota_{k-1}\, w))\ *\ (\iota_{k-1} w))\Big) * \pi_k\Big(p'(\iota_{k-1}\, w)\Big)\\
&\quad= \pi_k\Big(p((\iota^\circ_k p') \circ_k (\iota_{k-1}\, w))\Big) * \pi_k\Big(p'(\iota_{k-1}\, w)\Big)\\
&\quad= \pi_k\Big((p \otimes_k (\iota^\circ_k p'))(\iota_{k-1}\, w)\Big) * \pi_k\Big(p'(\iota_{k-1}\, w)\Big)\\
&\quad= \pi_k\Big((p \otimes_k (\iota^\circ_k p'))(\iota_{k-1}\, w) * p'(\iota_{k-1}\, w)\Big)\\
&\quad= \iota^\circ_k(p \otimes_k (\iota^\circ_k p')\ *\ p')(w).
\end{aligned}$$

Since this holds for any w, the sup metric on $Assert_k$ gives the required approximate equality $\iota^\circ_k(p \otimes_k (\iota^\circ_k p')\ *\ p') \overset{k}{=} (\iota^\circ_k\, p) \circ_k (\iota^\circ_k\, p')$.

For the proof of the second statement, let $w, w' \in W_k$, and let $w_0 \in \frac{1}{2}\cdot W_k$. By the assumption C6 that ι°_{k-1} distributes over $\circ_{k-1}$, we obtain the approximate equality

$$\iota^\circ_{k-1}(w' \circ_k w_0) \ \overset{k-1}{=}\ (\iota^\circ_{k-1}\, w') \circ_{k-1} (\iota^\circ_{k-1}\, w_0)$$

in W_{k-1}. Furthermore, by C2 that we have just shown,

$$\iota^\circ_k(\iota_k(w')) \ \overset{k}{=}\ w'$$

and hence the non-expansiveness of ι°_{k-1} and $\circ_k$ implies that

$$\iota^\circ_{k-1}(\iota^\circ_k(\iota_k(w')) \circ_k w_0) \ \overset{k}{=}\ \iota^\circ_{k-1}(w' \circ_k w_0).$$

If we combine what we have obtained so far,

$$\iota^\circ_{k-1}(\iota^\circ_k(\iota_k(w')) \circ_k w_0) \ \overset{k}{=}\ \iota^\circ_{k-1}(w' \circ_k w_0) \ \overset{k-1}{=}\ (\iota^\circ_{k-1}\, w') \circ_{k-1} (\iota^\circ_{k-1}\, w_0)$$

holds in W_{k-1}. Thus, in the scaled space $\frac{1}{2}\cdot W_{k-1}$, we have that

$$\iota^\circ_{k-1}(\iota^\circ_k(\iota_k(w')) \circ_k w_0) \ \overset{k}{=}\ (\iota^\circ_{k-1}\, w') \circ_{k-1} (\iota^\circ_{k-1}\, w_0).$$

Thus, using the definition of $\circ_k$ in terms of $\otimes_{k-1}$ and the lifted $*$,

$$\begin{aligned}
&((\iota_k\, w) \otimes_k (\iota_k^\circ(\iota_k(w'))) * (\iota_k\, w'))(w_0) \\
&\quad = (\iota_k\, w)((\iota_k^\circ(\iota_k(w'))) \circ_k w_0) * (\iota_k\, w')(w_0) \\
&\quad = \pi_{k+1}(w(\iota_{k-1}^\circ((\iota_k^\circ(\iota_k(w'))) \circ_k w_0))) * \pi_{k+1}(w'(\iota_{k-1}^\circ\, w_0)) \\
&\quad \overset{k}{=} \pi_{k+1}(w((\iota_{k-1}^\circ\, w') \circ_{k-1} (\iota_{k-1}^\circ\, w_0))) * \pi_{k+1}(w'(\iota_{k-1}^\circ\, w_0)) \\
&\quad = \pi_{k+1}((w \otimes_{k-1} (\iota_{k-1}^\circ\, w'))(\iota_{k-1}^\circ\, w_0)) * \pi_{k+1}(w'(\iota_{k-1}^\circ\, w_0)) \\
&\quad = \pi_{k+1}\Big((w \otimes_{k-1} (\iota_{k-1}^\circ\, w'))(\iota_{k-1}^\circ\, w_0) * w'(\iota_{k-1}^\circ\, w_0)\Big) \\
&\quad = \pi_{k+1}((w \circ_k w')(\iota_{k-1}^\circ\, w_0)) \\
&\quad = \iota_k(w \circ_k w')(w_0).
\end{aligned}$$

Since this approximate equality holds for all w_0, we obtain

$$(\iota_k\, w) \otimes_k (\iota_k^\circ(\iota_k w')) * (\iota_k\, w') \overset{k}{=} \iota_k(w \circ_k w')$$

by definition of the sup metric on $Assert_k$.

Finally, we prove the third statement. By the definition, for all $w_0 \in \frac{1}{2} \cdot W_{k-1}$,

$$(\iota_k^\circ(\lambda w.\mathsf{e}))(w_0) \;=\; \pi_k\Big((\lambda w.\mathsf{e})(\iota_{k-1}(w_0))\Big) \;=\; \pi_k(\mathsf{e}) \;=\; \mathsf{e} \;=\; e_k(w_0).$$

where the third equality follows from $k > 0$. We have just shown the third statement. □

C.2 Inverse limit construction

In this subsection we construct the inverse limit of the Cauchy tower $(W_k, \iota_k^\circ)_k$ described above, and we will show that it has the properties stated in Theorem 11. Let us define $\iota_{k,l} : W_k \to W_l$ and $\iota_{k,l}^\circ : W_l \to W_k$ for all $0 \le k < l$ as follows:

$$\begin{aligned}
\iota_{k,l} &= \iota_{l-1} \cdot \ldots \cdot \iota_{k+1} \cdot \iota_k \\
\iota_{k,l}^\circ &= \iota_k^\circ \cdot \iota_{k+1}^\circ \cdot \ldots \cdot \iota_{l-1}^\circ
\end{aligned}$$

Note that all ι_{kl} and ι_{kl}° are non-expansive, by Lemma 18.

We define an metric space W by

$$W \overset{def}{=} \Big\{ x \in \textstyle\prod_{k\ge 0} W_k \mid \forall k \ge 0.\ \iota_k^\circ(x_{k+1}) = x_k \Big\}$$

with distance $d_W(x,y) = \sup_k\ d_{W_k}(x_k, y_k)$. This makes W a complete ultrametric space, with limits of Cauchy chains given componentwise. Next, we equip W with an operation $\circ : W \times W \to W$, defined by

$$(x_k)_{k\ge 0} \circ (y_k)_{k\ge 0} \overset{def}{=} \big(\lim_{j>k} \iota_{k,j}^\circ(x_j \circ_j y_j)\big)_{k\ge 0}.$$

Note that the limits exist: $\iota^\circ_j(x_{j+1} \circ_{j+1} y_{j+1}) \overset{j}{=} \iota^\circ_j(x_{j+1}) \circ_j \iota^\circ_j(y_{j+1}) = x_j \circ_j y_j$ by Lemma 25, and so $(\iota^\circ_{k,j}(x_j \circ_j y_j))_{j>k}$ forms a Cauchy sequence by the non-expansiveness of $\iota^\circ_{k,j}$. Moreover, we have $\iota^\circ_k(\lim_{j>k+1} \iota^\circ_{k+1,j}(x_j \circ_j y_j)) = \lim_{j>k+1} \iota^\circ_k \iota^\circ_{k+1,j}(x_j \circ_j y_j)$ which shows $(x \circ y)_k = \iota^\circ_k((x \circ y)_{k+1})$, i.e., $\circ$ is well-defined.

We also define $e \in W$ by

$$e \overset{def}{=} (e_k)_{k\geq 0}$$

From C6, it follows that $\iota^\circ_k(e_{k+1}) = e_k$ for all k, so that e is indeed an element of W.

Lemma 26. *$(W, \circ, e)$ is a monoid with non-expansive multiplication $\circ$.*

Proof. By Lemma 21 and the definition of e and $\circ$ we have $e \circ x = x \circ e = x$ for all $x \in W$. To see the associativity of $\circ$ suppose $x, y, z \in W$. Lemma 20 shows for all j: $(x_j \circ_j y_j) \circ_j z_j \overset{j}{=} x_j \circ_j (y_j \circ_j z_j)$. We obtain:

$$\begin{aligned}
x_j \circ_j (y \circ z)_j &= x_j \circ_j (\lim_{l>j} \iota^\circ_{j,l}(y_l \circ_l z_l)) \\
&= \lim_{l>j} x_j \circ_j \iota^\circ_{j,l}(y_l \circ_l z_l)) && \text{(non-expansiveness of } \circ_j) \\
&\overset{j}{=} \lim_{l>j} x_j \circ_j (\iota^\circ_{j,l}(y_l) \circ_j \iota^\circ_{j,l}(z_l)) && \text{(Lemma 25)} \\
&= \lim_{l>j} x_j \circ_j (y_j \circ_j z_j) && (y, z \in W) \\
&\overset{j}{=} \lim_{l>j} (x_j \circ_j y_j) \circ_j z_j && \text{(Lemma 20)} \\
&\overset{j}{=} (x \circ y)_j \circ_j z_j.
\end{aligned}$$

Thus, for any $\varepsilon > 0$ there exists $n \geq 0$ sufficiently large such that

$$\forall j \geq n.\ d_{W_j}((x \circ y)_j \circ_j z_j,\ x_j \circ_j (y \circ z)_j) < \varepsilon.$$

Since $\iota^\circ_{k,j}$ is non-expansive, this yields

$$\begin{aligned}
\forall k.\ ((x \circ y) \circ z)_k &= \lim_{j>k} \iota^\circ_{k,j}((x \circ y)_j \circ_j z_j) \\
&= \lim_{j>k} \iota^\circ_{k,j}(x_j \circ_j (y \circ z)_j) \\
&= (x \circ (y \circ z))_k.
\end{aligned}$$

This proves $(x \circ y) \circ z = x \circ (y \circ z)$.

Finally, $\circ$ is non-expansive since each $\circ_j$ and $\iota^\circ_{k,j}$ is non-expansive. □

As shown in the preceding lemma, $\circ$ is associative and has e as a unit, therefore we can consider the induced preorder on W:

$$w \sqsubseteq w' \ \Leftrightarrow\ \exists w_0.\ w' = w \circ w_0$$

Note that if $w' = w \circ w''$ then $w'_k \stackrel{k}{=} w_k \circ_k w''_k$ for all k, and therefore we obtain:

$$w \sqsubseteq w' \;\Rightarrow\; \forall k.\; w_k \sqsubseteq_k w'_k. \tag{15}$$

It remains to establish an isomorphism $W \cong \frac{1}{2} \cdot W \to_m \mathit{UAdm}$ in *CBUlt* where the monotonicity refers to the induced preorder $\sqsubseteq$ on W. To this end, first note that for each k, each $(w_k)_{k\geq 0}$ and $(w'_k)_{k\geq 0}$ in W we have

$$\begin{aligned} w_{k+1}(w'_k) &= \iota^\circ_{k+1}(w_{k+2})(\iota^\circ_k\, w'_{k+1}) \\ &= (\pi_{k+1} \cdot w_{k+2} \cdot \iota_k \cdot \iota^\circ_k)(w'_{k+1}) \\ &\stackrel{k+1}{=} (w_{k+2} \cdot \iota_k \cdot \iota^\circ_k)(w'_{k+1}) \\ &\stackrel{k+1}{=} w_{k+2}(w'_{k+1}) \qquad \text{(C2 and scaling by 1/2).} \end{aligned}$$

Thus, $(\lambda w'.w_{k+1}(w'_k))_{k\geq 0}$ is a Cauchy sequence in $\frac{1}{2}\cdot W \to \mathit{UAdm}$. In fact, it is a sequence in the (complete) subspace of monotone maps, by (15) and the fact that each w_k is monotone, and therefore this sequence has a limit in $\frac{1}{2}\cdot W \to_m \mathit{UAdm}$. Accordingly we may set

$$\iota(w) \stackrel{def}{=} \lim_k (\lambda w' \in W.w_{k+1}(w'_k))$$

For $p \in \frac{1}{2} \cdot W \to_m \mathit{UAdm}$ we define $\iota^\circ(p)_k \in W_k$ by the following two cases:

$$\begin{aligned} \iota^\circ(p)_0 &\stackrel{def}{=} \langle\rangle \\ \iota^\circ(p)_{k+1} &\stackrel{def}{=} \lambda w \in \tfrac{1}{2}\cdot W_k.\, \pi_{k+1}(p(\big(\lim_{l>\max\{i,k\}}\; \iota^\circ_{i,l} \cdot \iota_{k,l}(w)\big)_{i\geq 0})) \end{aligned}$$

For this latter definition, one checks that $(\lim_{l>\max\{i,k\}}\; \iota^\circ_{i,l} \cdot \iota_{k,l}(w))_{i\geq 0}$ is an element of W, so that p can be applied. Next, $\iota^\circ(p)_{k+1}$ is monotone. To see this, let $w_1 \sqsubseteq_k w_2$, so there exists $w_0 \in W_k$ such that $w_2 \stackrel{k}{=} w_1 \circ_k w_0$ in W_k; we must show that $(\iota^\circ\, p)_{k+1}(w_1) \subseteq (\iota^\circ\, p)_{k+1}(w_2)$. Let $x_j = (\lim_{l>\max\{i,k\}}\; \iota^\circ_{i,l} \cdot \iota_{k,l}(w_j))_{i\geq 0}$ for $j = 0, 1, 2$. Then, $x_2 \stackrel{k}{=} x_1 \circ x_0$ in W. From the non-expansiveness and monotonicity of p it follows that $p(x_2) \stackrel{k+1}{=} p(x_1 \circ x_0) \supseteq p(x_1)$, and therefore $\pi_{k+1}(p(x_1)) \subseteq \pi_{k+1}(p(x_2))$, which yields the required monotonicity of $(\iota^\circ\, p)_{k+1}$. Finally, this definition of ι° satisfies $\iota^\circ_k((\iota^\circ\, p)_{k+1}) = (\iota^\circ\, p)_k$ for all k, and therefore $\iota^\circ\, p \in W$.

Lemma 27 (Non-expansiveness of ι° and ι). *$\iota^\circ : (\frac{1}{2}\cdot W \to_m \mathit{UAdm}) \to W$ and $\iota : W \to (\frac{1}{2}\cdot W \to_m \mathit{UAdm})$ are non-expansive.*

Proof. To see the non-expansiveness of ι, suppose $x \stackrel{n}{=} y$ in W. Thus, $x_k \stackrel{n}{=} y_k$ holds in $W_k = \frac{1}{2}\cdot W_{k-1} \to_m \mathit{UAdm}$, for all k, and therefore $\lambda w.x_k(w_{k-1}) \stackrel{n}{=} \lambda w.y_k(w_{k-1})$ holds in $\frac{1}{2}\cdot W \to_m \mathit{UAdm}$, for all k. Consequently, $\iota(x) \stackrel{n}{=} \iota(y)$ in $\frac{1}{2}\cdot W \to_m \mathit{UAdm}$.

To see the non-expansiveness of ι°, suppose $p \stackrel{n}{=} q$ in $\frac{1}{2}\cdot W \to_m UAdm$. We must show that $\iota^\circ(p)_j \stackrel{n}{=} \iota^\circ(q)_j$ for all $j \geq 0$. This is clear in the case $j = 0$, so assume $j > 0$. Let $k = j - 1$ and pick $w \in W_k$. Then by assumption we have

$$p(\big(\lim_{l>\max\{i,k\}}\ \iota^\circ_{i,l}\cdot\iota_{k,l}(w)\big)_i) \stackrel{n}{=} q(\big(\lim_{l>\max\{i,k\}}\ \iota^\circ_{i,l}\cdot\iota_{k,l}(w)\big)_i)$$

and therefore $\iota^\circ(p)_j \stackrel{n}{=} \iota^\circ(q)_j$ by the sup metric on W_j. □

Lemma 28. *$\iota^\circ\cdot\iota = id_W$ and $\iota\cdot\iota^\circ = id_{\frac{1}{2}\cdot W \to_m UAdm}$.*

Proof. We first verify that $\iota^\circ\cdot\iota = id_W$. We will show $(\iota^\circ(\iota\, w))_j \stackrel{j}{=} w_j$ for all j. This gives the desired equality, because

$$(\iota^\circ(\iota\, w))_l \;=\; \lim_{j>l}\iota^\circ_{l,j}\Big((\iota^\circ(\iota\, w))_j\Big), \qquad\qquad w_l \;=\; \lim_{j>l}\iota^\circ_{l,j}(w_j),$$

and $\iota^\circ_{j,l}$ is non-expansive for all k. Clearly $(\iota^\circ(\iota\, w))_0 \stackrel{0}{=} w_0$, and moreover:

$$\begin{aligned}
&(\iota^\circ(\iota\, w))_{j+1}\\
&= (\iota^\circ(\lim_k(\lambda w'\in W.w_{k+1}(w'_k))))_{j+1}\\
&= \lambda x\in W_j.\ \pi_{j+1}\big((\lim_k(\lambda w'\in W.w_{k+1}(w'_k))\ \big(\lim_{l>\max\{i,j\}}\ (\iota^\circ_{i,l}\cdot\iota_{j,l})(x)\big)_{i\geq 0}\big)\\
&\stackrel{j+1}{=} \lambda x\in W_j.\ (\lim_k(\lambda w'\in W.w_{k+1}(w'_k))\ \big(\lim_{l>\max\{i,j\}}\ (\iota^\circ_{i,l}\cdot\iota_{j,l})(x)\big)_{i\geq 0}\\
&= \lambda x\in W_j.\ \lim_k w_{k+1}\big(\lim_{l>\max\{k,j\}}\ (\iota^\circ_{k,l}\cdot\iota_{j,l})(x)\big)\\
&= \lambda x\in W_j.\ \lim_{k>j} w_{k+1}\big(\lim_{l>k}\ (\iota^\circ_{k,l}\cdot\iota_{j,l})(x)\big)\\
&= \lambda x\in W_j.\ \lim_{k>j} w_{k+1}\big(\lim_{l>k}\ (\iota^\circ_{k,l}\cdot\iota_{k,l}\cdot\iota_{j,k})(x)\big)\\
&= \lambda x\in W_j.\ \lim_{k>j} w_{k+1}\big(\lim_{l>k}\ (\iota_{j,k})(x)\big) \qquad\qquad (\text{since } \iota^\circ_{k,l}\cdot\iota_{k,l}\stackrel{k}{=} id_{W_k} \text{ by C2})\\
&= \lambda x\in W_j.\ \lim_{k>j} w_{k+1}(\iota_{j,k}(x))\\
&\stackrel{j+1}{=} \lambda x\in W_j.\ \lim_{k>j}(\pi_{j+1}\cdot\ldots\cdot\pi_k\cdot w_{k+1}\cdot\iota_{j,k})(x) \quad (\text{since } \pi_{j+1}\cdot\ldots\cdot\pi_k\stackrel{j+1}{=} id_{UAdm})\\
&= \lambda x\in W_j.\ \lim_{k>j}(\iota^\circ_{j+1,k})(w_{k+1})(x)\\
&= \lambda x\in W_j.\ \lim_{k>j} w_{j+1}(x)\\
&= \lambda x\in W_j.\ w_{j+1}(x)\\
&= w_{j+1}.
\end{aligned}$$

To see that $\iota\cdot\iota^\circ = id_{\frac{1}{2}\cdot W\to_m UAdm}$ let $p\in\frac{1}{2}\cdot W\to_m UAdm$ and $w\in W$; we show $\iota(\iota^\circ\, p)(w) = p(w)$. Since $w = (w_k)_{k\geq 0}\in W$ we have $w_k = \iota^\circ_k(w_{k+1})$ for all k, and thus

$$\iota_{k,l}w_k \;=\; \iota_{k,l}(\iota^\circ_{k,l}\, w_l) \;\stackrel{k}{=}\; w_l$$

for all k and all $l > k$. For fixed i this yields

$$\lim_{l>\max\{i,k\}} \iota^{\circ}_{i,l} \cdot \iota_{k,l}(w_k) \overset{k}{=} \lim_{l>\max\{i,k\}} \iota^{\circ}_{i,l}(w_l)$$

and therefore

$$\forall i.\ \lim_k \Big(\lim_{l>\max\{i,k\}} \iota^{\circ}_{i,l} \cdot \iota_{k,l}(w_k) \Big) = \lim_k \Big(\lim_{l>\max\{i,k\}} \iota^{\circ}_{i,l}(w_l) \Big) = \lim_{l>i} \iota^{\circ}_{i,l}(w_l) = w_i \tag{16}$$

We calculate

$$\begin{aligned}
\iota(\iota^\circ\, p)(w) &= \Big(\lim_k \Big(\lambda w' \in W.\, (\iota^\circ\, p)_{k+1}\, w'_k\Big)\Big)(w) \\
&= \lim_k \left((\iota^\circ\, p)_{k+1}\, w_k\right) && \text{(limits are pointwise)} \\
&= \lim_k \Big((\pi_{k+1} \cdot p)\Big(\Big(\lim_{l>\max\{i,k\}} \iota^{\circ}_{i,l} \cdot \iota_{k,l}(w_k)\Big)_{i\geq 0}\Big)\Big) \\
&= \lim_k \Big(p\Big(\Big(\lim_{l>\max\{i,k\}} \iota^{\circ}_{i,l} \cdot \iota_{k,l}(w_k)\Big)_{i\geq 0}\Big)\Big) && \text{(since } \pi_{k+1} \overset{k+1}{=} id_{\,UAdm}\text{)} \\
&= p\Big(\lim_k \Big(\Big(\lim_{l>\max\{i,k\}} \iota^{\circ}_{i,l} \cdot \iota_{k,l}(w_k)\Big)_{i\geq 0}\Big)\Big) && (p \text{ non-expansive}) \\
&= p\Big(\Big(\lim_k \lim_{l>\max\{i,k\}} \iota^{\circ}_{i,l} \cdot \iota_{k,l}(w_k)\Big)_{i\geq 0}\Big) && \text{(limits are pointwise)} \\
&= p((w_i)_{i\geq 0}) && \text{(by (16))} \\
&= p(w)
\end{aligned}$$

This concludes the proof. □

Finally, we check that we have the desired relationship between $\circ$, $\otimes$ and the isomorphism:

Lemma 29. *For all $w, w' \in W$, $w \circ w' = \iota^\circ(\iota(w) \otimes w' * \iota(w'))$, or equivalently:*

$$(p \otimes (\iota^\circ\, q) * q)(w) = (\iota(\iota^\circ(p) \circ \iota^\circ(q)))(w)$$

for all $p, q \in Assert = \frac{1}{2} \cdot W \to_m UAdm$ and all $w \in W$.

Proof. First note that for all $r \in \mathit{Assert}$ and $w \in W$ the following equation holds:

$$
\begin{aligned}
&\lim_k \lim_{j>k+1} (\iota^\circ\, r)_j(\iota_{k,j-1}\, w_k) \\
&= \lim_k \lim_{j>k+1} (\pi_j \cdot r)\Big(\Big(\lim_{l>\max\{i,j\}} \iota^\circ_{i,l} \cdot \iota_{j,l}(\iota_{k,j-1}\, w_k)\Big)_{i\geq 0}\Big) \\
&= \lim_k \lim_{j>k+1} r\Big(\Big(\lim_{l>\max\{i,j\}} \iota^\circ_{i,l} \cdot \iota_{j,l}(\iota_{k,j-1}\, w_k)\Big)_{i\geq 0}\Big) \qquad (\text{since } \pi_j \overset{j}{=} id_{UAdm}) \\
&= r\Big(\lim_k \lim_{j>k+1} \big(\lim_{l>\max\{i,j\}} \iota^\circ_{i,l} \cdot \iota_{j,l}(\iota_{k,j-1}\, w_k)\big)_{i\geq 0}\Big) \quad (\text{since } r \text{ is non-expansive}) \\
&= r\Big(\lim_k \lim_{j>k+1} \big(\lim_{l>\max\{i,j\}} (\iota^\circ_{i,l} \cdot \iota_{k,l})(w_k)\big)_{i\geq 0}\Big) \\
&= r\Big(\Big(\lim_k \lim_{j>k+1} \lim_{l>\max\{i,j\}} (\iota^\circ_{i,l} \cdot \iota_{k,l})(w_k)\Big)_{i\geq 0}\Big) \qquad (\text{limits are pointwise}) \\
&= r\Big(\big(\lim_k \lim_{l>\max\{i,k+2\}} (\iota^\circ_{i,l} \cdot \iota_{k,l})(w_k)\big)_{i\geq 0}\Big) \\
&= r\Big(\big(\lim_{k>i} \lim_{l>k+2} (\iota^\circ_{i,l} \cdot \iota_{k,l})(w_k)\big)_{i\geq 0}\Big) \\
&= r\Big(\big(\lim_{k>i} \lim_{l>k+2} (\iota^\circ_{i,k}\, w_k)\big)_{i\geq 0}\Big) \qquad (\text{since } \iota^\circ_{i,l} \cdot \iota_{k,l} \overset{k}{=} \iota^\circ_{i,k} \text{ by C2}) \\
&= r\Big(\big(\lim_{k>i} (\iota^\circ_{i,k}\, w_k)\big)_{i\geq 0}\Big) \\
&= r\Big(\big(\lim_{k>i} w_i\big)_{i\geq 0}\Big) \\
&= r(w).
\end{aligned}
$$

Now let $p, q \in \mathit{Assert}$ and $w \in W$. By definition of $\otimes$ we have

$$(p \otimes (\iota^\circ\, q) \ast q)(w) = p(\iota^\circ\, q \circ w) * q(w),$$

and we must show that this equals $\iota((\iota^\circ(p) \circ \iota^\circ(q)))(w)$:

$$
\begin{aligned}
&\iota(\iota^\circ(p) \circ \iota^\circ(q))(w) \\
&= \iota\Big(\Big(\lim_{j>k}\ \iota^\circ_{k,j}((\iota^\circ p)_j \circ_j (\iota^\circ q)_j)\Big)_{k\geq 0}\Big)(w) \\
&= \lim_k \Big(\lim_{j>k+1}\ \iota^\circ_{k+1,j}((\iota^\circ p)_j \circ_j (\iota^\circ q)_j)\Big)(w_k) \\
&= \lim_k \Big(\lim_{j>k+1}\ \iota^\circ_{k+1,j}((\iota^\circ p)_j \circ_j (\iota^\circ q)_j)(w_k)\Big) \\
&= \lim_k \lim_{j>k+1} \Big((\pi_{k+1} \cdot \ldots \cdot \pi_{j-1} \cdot ((\iota^\circ p)_j \circ_j (\iota^\circ q)_j) \cdot \iota_{k,j-1})(w_k)\Big) \\
&= \lim_k \lim_{j>k+1} \Big(((\iota^\circ p)_j \circ_j (\iota^\circ q)_j) \cdot \iota_{k,j-1})(w_k)\Big) \\
&= \lim_k \lim_{j>k+1} \Big(((\iota^\circ p)_j \circ_j (\iota^\circ q)_j)(\iota_{k,j-1}(w_k))\Big) \\
&= \lim_k \lim_{j>k+1} \Big(\Big((\iota^\circ p)_j \otimes_{j-1} \iota^\circ_{j-1}((\iota^\circ q)_j)\Big)(\iota_{k,j-1}(w_k))\ *\ (\iota^\circ q)_j(\iota_{k,j-1}(w_k))\Big) \\
&= \Big(\lim_k \lim_{j>k+1} \Big((\iota^\circ p)_j \otimes_{j-1} \iota^\circ_{j-1}((\iota^\circ q)_j)\Big)(\iota_{k,j-1}(w_k))\Big)\ *\ \Big(\lim_k \lim_{j>k+1} (\iota^\circ q)_j(\iota_{k,j-1}(w_k))\Big) \\
&= \lim_k \lim_{j>k+1} \Big((\iota^\circ p)_j \otimes_{j-1} \iota^\circ_{j-1}((\iota^\circ q)_j)\Big)(\iota_{k,j-1}(w_k))\ *\ q(w) \\
&= \lim_k \lim_{j>k+1} (\iota^\circ p)_j\Big(\iota^\circ_{j-1}((\iota^\circ q)_j) \circ_{j-1} (\iota_{k,j-1}(w_k))\Big)\ *\ q(w) \\
&= \lim_k \lim_{j>k+1} (\iota^\circ p)_j\Big((\iota^\circ q)_{j-1} \circ_{j-1} (\iota_{k,j-1}(w_k))\Big)\ *\ q(w) \\
&= \lim_k \lim_{j>k+1} (\iota^\circ p)_j\Big((\iota^\circ q)_{j-1} \circ_{j-1} (\iota_{k,j-1} \cdot \iota^\circ_{k,j-1})(w_{j-1})\Big)\ *\ q(w) \\
&= \lim_k \lim_{j>k+1} (\iota^\circ p)_j\Big((\iota^\circ q)_{j-1} \circ_{j-1} w_{j-1})\Big)\ *\ q(w) \qquad\qquad (\text{by } \iota_{k,j-1} \cdot \iota^\circ_{k,j-1} \overset{k}{=} id) \\
&= \lim_k \lim_{j>k+1} (\iota^\circ p)_j\Big(\lim_{l>j-1}\ \iota^\circ_{j-1,l}((\iota^\circ q)_l) \circ_{j-1} \iota^\circ_{j-1,l}(w_l)\Big)\ *\ q(w) \\
&= \lim_k \lim_{j>k+1} (\iota^\circ p)_j\Big(\lim_{l>j-1}\ \iota^\circ_{j-1,l}((\iota^\circ q)_l \circ_l w_l)\Big) * q(w)\ \ (\text{by } \iota^\circ_{a,b}(x \circ_b y) \overset{a}{=} \iota^\circ_{a,b}(x) \circ_a \iota^\circ_{a,b}(y)) \\
&= \Big(\lim_k \lim_{j>k+1} (\iota^\circ p)_j\big(((\iota^\circ q) \circ w)_{j-1}\big)\Big)\ *\ q(w) \\
&= \Big(\lim_k \lim_{j>k+1} (\iota^\circ p)_j\big((\iota_{k,j-1} \cdot \iota^\circ_{k,j-1})((\iota^\circ q) \circ w)_{j-1}\big)\Big) * q(w)\ \ (\text{by } \iota_{k,j-1} \cdot \iota^\circ_{k,j-1} \overset{k}{=} id) \\
&= \Big(\lim_k \lim_{j>k+1} (\iota^\circ p)_j\big(\iota_{k,j-1}((\iota^\circ q) \circ w)_k\big)\Big)\ *\ q(w) \\
&= p((\iota^\circ q) \circ w)\ *\ q(w)
\end{aligned}
$$

where the ninth and the last equation are by the equality established above. □

D A concrete model with non-recursive worlds

In this appendix, we describe a model based on our framework in Sections 2 and 3, where only world-independent assertions can be used as invariants. This model has two interesting features. First, the set W of worlds has a simple definition, which does not involve solving recursive domain equations. Second, by restricting possible invariants, the model allows us to have versions of anti-frame and frame rules that are stronger than those in Section 3. Our description will focus on these two features.

Let *Heap* be a pointed cpo of heaps that satisfies the conditions in Section 2 (not necessarily the same heap model as in Section 5). Assume that *Heap* contains the unit, denoted e_h, for the heap combining operation $h \cdot h'$, such that $\pi_0(e_h) = \bot$ and $\pi_k(e_h) = e_h$ for all $k > 0$. Furthermore, we assume that $\bot \cdot h = h \cdot \bot = \bot$. The operation $\cdot$ and its unit e_h give rise to a monoid structure on the set *UAdm* of uniform admissible subsets of *Heap*. The monoid composition is the usual separating conjunction $p * q$ (described in Section 2) and the unit I is the set $\{\bot, e_h\}$.

The model we are about to describe is an instance of our semantic framework where the monoid $(W, e, \circ)$ of worlds and the coercion function are defined as follows:

$$(W, e, \circ) \stackrel{def}{=} (UAdm, I, *), \qquad i(w) \stackrel{def}{=} \lambda w_0.\, w.$$

Note that the definition is simple, without involving the solutions of any recursive domain equations. Intuitively, this definition means that invariants are assertions not depending on worlds, as formalized by the fact that $i(w)$ is a constant function. The defined worlds W and coercion map i satisfy the requirements of our framework.

Lemma 30. *The worlds W and the coercion i just defined satisfy the healthiness conditions in Section 2.*

Proof. Condition 1 requires the preservation of $- \circ w_0$ by the coercion function i. We can show the condition by simply unfolding the definitions of i and $\circ$. Concretely, for all $w, w_0, w_1 \in W$,

$$\begin{array}{lll} i(w \circ w_0)(w_1) & = & w \circ w_0 \qquad\qquad\qquad\qquad\quad (\text{by def. of } i) \\ & = & w * w_0 \qquad\qquad\qquad\qquad\quad (\text{by def. of } w \circ w_0) \\ & = & i(w)(w_0 \circ w_1) * i(w_0)(w_1) \qquad (\text{by def. of } i) \\ & = & (i(w) \circ w_0)(w_1). \qquad\qquad\quad (\text{by def. of } i(w) \circ w_0) \end{array}$$

For Condition 2, consider worlds w_0, w_1 in W. We claim that (w_0, w_1) is a commutative pair for (w_0, w_1). Since $*$ is commutative, we can prove the first requirement of being a commutative pair as follows:

$$w_0 \circ w_1 \;=\; w_0 * w_1 \;=\; w_1 * w_0 \;=\; w_1 \circ w_0.$$

The other two requirements for a commutative pair follow from the fact that $i(w_0)$ and $i(w_1)$ are constant functions:

$$(i(w_0)\otimes w_1)(w) \;=\; i(w_0)(w_1\circ w) \;=\; w_0, \quad (i(w_1)\otimes w_0)(w) \;=\; i(w_1)(w_0\circ w) \;=\; w_1.$$

□

By the results of Section 3, this model validates the anti-frame and frame rules in Fig. 1. But, in fact, it validates even stronger versions of those rules, expressed in the proposition below:

Proposition 31. *The following* axiom *variants of the anti-frame rule and the deep frame rule are sound:*

$$\textsc{Anti-frame Axiom}\quad \frac{}{\{p\}c\{q\circ w_0\} \models \{p\}c\{q\}} \qquad \textsc{Deep-frame Axiom}\quad \frac{}{\{p\}c\{q\} \models \{p\circ w_0\}c\{q\circ w_0\}}$$

Proof. The main idea of the proof is to exploit the commutativity of the monoid operation $\circ$ for worlds. Since this $\circ$ operation is already associative, commutativity means that we can freely exchange the order of world compositions by the operation. In other instances of our framework commutativity does not normally hold for the monoid operation for worlds, and the instances do not validate the axioms in this proposition.

To prove the soundness of the anti-frame axiom, consider worlds w, w_1 and index k. We need to prove that

$$tri(p\circ(w\circ w_1),\ \pi_k(c),\ q\circ(w\circ w_1))$$

holds. We do this below using the assumption and the commutativity of the monoid operation for worlds:

$$\begin{array}{rcl} w \models_k \{p\}c\{q\circ w_0\} & \Rightarrow & tri(p\circ(w\circ w_1),\ \pi_k(c),\ (q\circ w_0)\circ(w\circ w_1)) \\ & \Rightarrow & tri(p\circ(w\circ w_1),\ \pi_k(c),\ q\circ(w_0\circ(w\circ w_1))) \\ & \Rightarrow & tri(p\circ(w\circ w_1),\ \pi_k(c),\ q\circ((w\circ w_1)\circ w_0)) \\ & \Rightarrow & tri(p\circ(w\circ w_1),\ \pi_k(c),\ (q\circ w\circ w_1)\circ w_0) \\ & \Rightarrow & tri(p\circ(w\circ w_1),\ \pi_k(c),\ (q\circ w\circ w_1)). \end{array}$$

The first implication follows from the definition of the validity of triples, and the second and fourth hold because of Lemma 2. The third implication uses the commutativity of the monoid operation $\circ$. The last implication holds because of Lemma 5.

Next, we move on to the deep-frame axiom. Pick worlds w, w_1 and index k. We need to prove that

$$tri((p\circ w_0)\circ(w\circ w_1),\ \pi_k(c),\ (q\circ w_0)\circ(w\circ w_1))$$

holds. We do this, again using the commutativity of the $\circ$ operation:

$$\begin{aligned} w \models_k \{p\}c\{q\} &\Rightarrow tri(p \circ (w \circ (w_0 \circ w_1)),\ \pi_k(c),\ q \circ (w \circ (w_0 \circ w_1))) \\ &\Rightarrow tri(p \circ (w_0 \circ (w \circ w_1)),\ \pi_k(c),\ q \circ (w_0 \circ (w \circ w_1))) \\ &\Rightarrow tri((p \circ w_0) \circ (w \circ w_1),\ \pi_k(c),\ (q \circ w_0) \circ (w \circ w_1)). \end{aligned}$$

The first implication follows from the validity of triples, and the second one uses the commutativity and associativity of the $\circ$ operation for worlds. The last implication holds because of Lemma 2. $\square$