The Iris 2.0 Documentation

August 24, 2016

Contents

1 Algebraic Structures

1.1 COFE

The model of Iris lives in the category of *Complete Ordered Families of Equivalences* (COFEs). This definition varies slightly from the original one in [2].

Definition 1 (Chain)**.** *Given some set T and an indexed family $(\stackrel{n}{=} \subseteq T \times T)_{n\in\mathbb{N}}$ of equivalence relations, a* chain *is a function $c : \mathbb{N} \to T$ such that $\forall n, m.\, n \leq m \Rightarrow c(m) \stackrel{n}{=} c(n)$.*

Definition 2. *A* complete ordered family of equivalences *(COFE) is a tuple $(T, (\stackrel{n}{=} \subseteq T \times T)_{n\in\mathbb{N}}, \lim : \mathrm{chain}(T) \to T)$ satisfying*

$$\forall n.\, (\stackrel{n}{=}) \text{ is an equivalence relation} \tag{cofe-equiv}$$
$$\forall n, m.\, n \geq m \Rightarrow (\stackrel{n}{=}) \subseteq (\stackrel{m}{=}) \tag{cofe-mono}$$
$$\forall x, y.\, x = y \Leftrightarrow (\forall n.\, x \stackrel{n}{=} y) \tag{cofe-limit}$$
$$\forall n, c.\, \lim(c) \stackrel{n}{=} c(n) \tag{cofe-compl}$$

The key intuition behind COFEs is that elements x and y are n-equivalent, notation $x \stackrel{n}{=} y$, if they are *equivalent for n steps of computation*, *i.e.,* if they cannot be distinguished by a program running for no more than n steps. In other words, as n increases, $\stackrel{n}{=}$ becomes more and more refined (cofe-mono)—and in the limit, it agrees with plain equality (cofe-limit). In order to solve the recursive domain equation in §6 it is also essential that COFEs are *complete*, *i.e.,* that any chain has a limit (cofe-compl).

Definition 3. *An element $x \in T$ of a COFE is called* discrete *if*

$$\forall y \in T.\, x \stackrel{0}{=} y \Rightarrow x = y$$

A COFE A is called discrete *if all its elements are discrete. For a set X, we write ΔX for the discrete COFE with $x \stackrel{n}{=} x' \triangleq x = x'$*

Definition 4. *A function $f : T \to U$ between two COFEs is* non-expansive *(written $f : T \xrightarrow{ne} U$) if*

$$\forall n, x \in T, y \in T.\, x \stackrel{n}{=} y \Rightarrow f(x) \stackrel{n}{=} f(y)$$

It is contractive *if*

$$\forall n, x \in T, y \in T.\, (\forall m < n.\, x \stackrel{m}{=} y) \Rightarrow f(x) \stackrel{n}{=} f(y)$$

Intuitively, applying a non-expansive function to some data will not suddenly introduce differences between seemingly equal data. Elements that cannot be distinguished by programs within n steps remain indistinguishable after applying f. The reason that contractive functions are interesting is that for every contractive $f : T \to T$ with T inhabited, there exists a *unique* fixed-point $\mathit{fix}(f)$ such that $\mathit{fix}(f) = f(\mathit{fix}(f))$.

Definition 5. *The category $\mathcal{COFE}$ consists of COFEs as objects, and non-expansive functions as arrows.*

Note that $\mathcal{COFE}$ is cartesian closed. In particular:

Definition 6. *Given two COFEs T and U, the set of non-expansive functions $\left\{f : T \xrightarrow{ne} U\right\}$ is itself a COFE with*

$$f \stackrel{n}{=} g \triangleq \forall x \in T.\, f(x) \stackrel{n}{=} g(x)$$

Definition 7. *A (bi)functor $F : \mathcal{COFE} \to \mathcal{COFE}$ is called* locally non-expansive *if its action F_1 on arrows is itself a non-expansive map. Similarly, F is called* locally contractive *if F_1 is a contractive map.*

The function space $(-) \xrightarrow{\mathrm{ne}} (-)$ is a locally non-expansive bifunctor. Note that the composition of non-expansive (bi)functors is non-expansive, and the composition of a non-expansive and a contractive (bi)functor is contractive. The reason contractive (bi)functors are interesting is that by America and Rutten's theorem [1, 3], they have a unique[1] fixed-point.

1.2 RA

Definition 8. *A* resource algebra *(RA) is a tuple* $(M, \mathcal{V} \subseteq M, |-| : M \to M^?, (\cdot) : M \times M \to M)$ *satisfying:*

$$\forall a, b, c.\ (a \cdot b) \cdot c = a \cdot (b \cdot c) \qquad \text{(RA-ASSOC)}$$

$$\forall a, b.\ a \cdot b = b \cdot a \qquad \text{(RA-COMM)}$$

$$\forall a.\ |a| \in M \Rightarrow |a| \cdot a = a \qquad \text{(RA-CORE-ID)}$$

$$\forall a.\ |a| \in M \Rightarrow ||a|| = |a| \qquad \text{(RA-CORE-IDEM)}$$

$$\forall a, b.\ |a| \in M \wedge a \preccurlyeq b \Rightarrow |b| \in M \wedge |a| \preccurlyeq |b| \qquad \text{(RA-CORE-MONO)}$$

$$\forall a, b.\ (a \cdot b) \in \mathcal{V} \Rightarrow a \in \mathcal{V} \qquad \text{(RA-VALID-OP)}$$

$$\textit{where} \qquad M^? \triangleq M \uplus \{\top\} \qquad\qquad a^? \cdot \top \triangleq \top \cdot a^? \triangleq a^?$$

$$a \preccurlyeq b \triangleq \exists c \in M.\ b = a \cdot c \qquad \text{(RA-INCL)}$$

RAs are closely related to *Partial Commutative Monoids* (PCMs), with two key differences:

1. The composition operation on RAs is total (as opposed to the partial composition operation of a PCM), but there is a specific subset $\mathcal{V}$ of *valid* elements that is compatible with the composition operation (RA-VALID-OP).

   This take on partiality is necessary when defining the structure of *higher-order* ghost state, CMRAs, in the next subsection.

2. Instead of a single unit that is an identity to every element, we allow for an arbitrary number of units, via a function $|-|$ assigning to an element a its *(duplicable) core* $|a|$, as demanded by RA-CORE-ID. We further demand that $|-|$ is idempotent (RA-CORE-IDEM) and monotone (RA-CORE-MONO) with respect to the *extension order*, defined similarly to that for PCMs (RA-INCL).

   Notice that the domain of the core is $M^?$, a set that adds a dummy element $\top$ to M. Thus, the core can be *partial*: not all elements need to have a unit. We use the metavariable $a^?$ to indicate elements of $M^?$. We also lift the composition $(\cdot)$ to $M^?$. Partial cores help us to build interesting composite RAs from smaller primitives.

   Notice also that the core of an RA is a strict generalization of the unit that any PCM must provide, since $|-|$ can always be picked as a constant function.

Definition 9. *It is possible to do a* frame-preserving update *from* $a \in M$ *to* $B \subseteq M$*, written* $a \rightsquigarrow B$*, if*

$$\forall a_{\mathrm{f}}^? \in M^?.\ a \cdot a_{\mathrm{f}}^? \in \mathcal{V} \Rightarrow \exists b \in B.\ b \cdot a_{\mathrm{f}}^? \in \mathcal{V}$$

We further define $a \rightsquigarrow b \triangleq a \rightsquigarrow \{b\}$*.*

The assertion $a \rightsquigarrow B$ says that every element $a_{\mathrm{f}}^?$ compatible with a (we also call such elements *frames*), must also be compatible with some $b \in B$. Notice that $a_{\mathrm{f}}^?$ could be $\top$, so the frame-preserving update can also be applied to elements that have *no* frame. Intuitively, this means that whatever assumptions the rest of the program is making about the state of γ, if these assumptions are compatible with a, then updating to b will not invalidate any of these assumptions. Since Iris ensures that the global ghost state is valid, this means that we can soundly update the ghost state from a to a non-deterministically picked $b \in B$.

[1] Uniqueness is not proven in Coq.

1.3 CMRA

Definition 10. *A* CMRA *is a tuple* $(M : \mathcal{COFE}, (\mathcal{V}_n \subseteq M)_{n\in\mathbb{N}},$
$|-| : M \xrightarrow{ne} M^?, (\cdot) : M \times M \xrightarrow{ne} M)$ *satisfying:*

$$\forall n, a, b.\, a \overset{n}{=} b \wedge a \in \mathcal{V}_n \Rightarrow b \in \mathcal{V}_n \tag{cmra-valid-ne}$$
$$\forall n, m.\, n \geq m \Rightarrow \mathcal{V}_n \subseteq \mathcal{V}_m \tag{cmra-valid-mono}$$
$$\forall a, b, c.\, (a \cdot b) \cdot c = a \cdot (b \cdot c) \tag{cmra-assoc}$$
$$\forall a, b.\, a \cdot b = b \cdot a \tag{cmra-comm}$$
$$\forall a.\, |a| \in M \Rightarrow |a| \cdot a = a \tag{cmra-core-id}$$
$$\forall a.\, |a| \in M \Rightarrow ||a|| = |a| \tag{cmra-core-idem}$$
$$\forall a, b.\, |a| \in M \wedge a \preccurlyeq b \Rightarrow |b| \in M \wedge |a| \preccurlyeq |b| \tag{cmra-core-mono}$$
$$\forall n, a, b.\, (a \cdot b) \in \mathcal{V}_n \Rightarrow a \in \mathcal{V}_n \tag{cmra-valid-op}$$
$$\begin{aligned}\forall n, a, b_1, b_2.\, & a \in \mathcal{V}_n \wedge a \overset{n}{=} b_1 \cdot b_2 \Rightarrow \\ & \exists c_1, c_2.\, a = c_1 \cdot c_2 \wedge c_1 \overset{n}{=} b_1 \wedge c_2 \overset{n}{=} b_2\end{aligned} \tag{cmra-extend}$$

where

$$a \preccurlyeq b \triangleq \exists c.\, b = a \cdot c \tag{cmra-incl}$$
$$a \overset{n}{\preccurlyeq} b \triangleq \exists c.\, b \overset{n}{=} a \cdot c \tag{cmra-inclN}$$

This is a natural generalization of RAs over COFEs. All operations have to be non-expansive, and the validity predicate $\mathcal{V}$ can now also depend on the step-index. We define the plain $\mathcal{V}$ as the "limit" of the $\mathcal{V}_n$:

$$\mathcal{V} \triangleq \bigcap_{n\in\mathbb{N}} \mathcal{V}_n$$

The extension axiom (cmra-extend). Notice that the existential quantification in this axiom is *constructive*, *i.e.,* it is a sigma type in Coq. The purpose of this axiom is to compute a_1, a_2 completing the following square:

$$\begin{array}{ccc} a & \overset{n}{=} & b \\ \| & & \| \\ a_1 \cdot a_2 & \overset{n}{=} & b_1 \cdot b_2 \end{array}$$

where the n-equivalence at the bottom is meant to apply to the pairs of elements, *i.e.,* we demand $a_1 \overset{n}{=} b_1$ and $a_2 \overset{n}{=} b_2$. In other words, extension carries the decomposition of b into b_1 and b_2 over the n-equivalence of a and b, and yields a corresponding decomposition of a into a_1 and a_2. This operation is needed to prove that $\triangleright$ commutes with separating conjunction:

$$\triangleright(P * Q) \Leftrightarrow \triangleright P * \triangleright Q$$

Definition 11. *An element ε of a CMRA M is called the* unit *of M if it satisfies the following conditions:*

1. *ε is valid:*
   $\forall n.\, \varepsilon \in \mathcal{V}_n$
2. *ε is a left-identity of the operation:*
   $\forall a \in M.\, \varepsilon \cdot a = a$
3. *ε is a discrete COFE element*
4. *ε is its own core:*
   $|\varepsilon| = \varepsilon$

Lemma 1. *If M has a unit ε, then the core $|-|$ is total,* i.e., $\forall a.\, |a| \in M$.

Definition 12. *It is possible to do a* frame-preserving update *from $a \in M$ to $B \subseteq M$, written $a \rightsquigarrow B$, if*

$$\forall n, a_{\mathrm{f}}^?.\, a \cdot a_{\mathrm{f}}^? \in \mathcal{V}_n \Rightarrow \exists b \in B.\, b \cdot a_{\mathrm{f}}^? \in \mathcal{V}_n$$

We further define $a \rightsquigarrow b \triangleq a \rightsquigarrow \{b\}$.

Note that for RAs, this and the RA-based definition of a frame-preserving update coincide.

Definition 13. *A CMRA M is* discrete *if it satisfies the following conditions:*

1. *M is a discrete COFE*
2. *$\mathcal{V}$ ignores the step-index:*
   $\forall a \in M.\, a \in \mathcal{V}_0 \Rightarrow \forall n, a \in \mathcal{V}_n$

Note that every RA is a discrete CMRA, by picking the discrete COFE for the equivalence relation. Furthermore, discrete CMRAs can be turned into RAs by ignoring their COFE structure, as well as the step-index of $\mathcal{V}$.

Definition 14. *A function $f : M_1 \to M_2$ between two CMRAs is* monotone *(written $f : M_1 \xrightarrow{mon} M_2$) if it satisfies the following conditions:*

1. *f is non-expansive*
2. *f preserves validity:*
   $\forall n, a \in M_1.\, a \in \mathcal{V}_n \Rightarrow f(a) \in \mathcal{V}_n$
3. *f preserves CMRA inclusion:*
   $\forall a \in M_1, b \in M_1.\, a \preccurlyeq b \Rightarrow f(a) \preccurlyeq f(b)$

Definition 15. *The category $\mathcal{CMRA}$ consists of CMRAs as objects, and monotone functions as arrows.*

Note that every object/arrow in $\mathcal{CMRA}$ is also an object/arrow of $\mathcal{COFE}$. The notion of a locally non-expansive (or contractive) bifunctor naturally generalizes to bifunctors between these categories.

2 COFE constructions

2.1 Next (type-level later)

Given a COFE T, we define $\blacktriangleright T$ as follows (using a datatype-like notation to define the type):

$$\blacktriangleright T \triangleq \mathsf{next}(x : T)$$
$$\mathsf{next}(x) \overset{n}{=} \mathsf{next}(y) \triangleq n = 0 \lor x \overset{n-1}{=} y$$

Note that in the definition of the carrier $\blacktriangleright T$, next is a constructor (like the constructors in Coq), *i.e.,* this is short for $\{\mathsf{next}(x) \mid x \in T\}$.

$\blacktriangleright(-)$ is a locally *contractive* functor from $\mathcal{COFE}$ to $\mathcal{COFE}$.

2.2 Uniform Predicates

Given a CMRA M, we define the COFE $\mathit{UPred}(M)$ of *uniform predicates* over M as follows:

$$\mathit{UPred}(M) \triangleq \left\{ \varphi : \mathbb{N} \times M \rightarrow \mathit{Prop} \;\middle|\; \begin{array}{l} (\forall n, x, y.\, \varphi(n, x) \land x \overset{n}{=} y \Rightarrow \varphi(n, y)) \land \\ (\forall n, m, x, y.\, \varphi(n, x) \land x \preccurlyeq y \land m \leq n \land y \in \mathcal{V}_m \Rightarrow \varphi(m, y)) \end{array} \right\}$$

where Prop is the set of meta-level propositions, *e.g.,* Coq's `Prop`. $\mathit{UPred}(-)$ is a locally non-expansive functor from $\mathcal{CMRA}$ to $\mathcal{COFE}$.

One way to understand this definition is to re-write it a little. We start by defining the COFE of *step-indexed propositions*: For every step-index, the proposition either holds or does not hold.

$$\begin{aligned} \mathit{SProp} &\triangleq \wp^{\downarrow}(\mathbb{N}) \\ &\triangleq \{X \in \wp(\mathbb{N}) \mid \forall n, m.\, n \geq m \Rightarrow n \in X \Rightarrow m \in X\} \\ X \overset{n}{=} Y &\triangleq \forall m \leq n.\, m \in X \Leftrightarrow m \in Y \end{aligned}$$

Notice that this notion of SProp is already hidden in the validity predicate $\mathcal{V}_n$ of a CMRA: We could equivalently require every CMRA to define $\mathcal{V}_{-}(-) : M \xrightarrow{\text{ne}} \mathit{SProp}$, replacing cmra-valid-ne and cmra-valid-mono.

Now we can rewrite $\mathit{UPred}(M)$ as monotone step-indexed predicates over M, where the definition of a "monotone" function here is a little funny.

$$\begin{aligned} \mathit{UPred}(M) &\cong M \xrightarrow{\text{mon}} \mathit{SProp} \\ &\triangleq \left\{ \varphi : M \xrightarrow{\text{ne}} \mathit{SProp} \;\middle|\; \forall n, m, x, y.\, n \in \varphi(x) \land x \preccurlyeq y \land m \leq n \land y \in \mathcal{V}_m \Rightarrow m \in \varphi(y) \right\} \end{aligned}$$

The reason we chose the first definition is that it is easier to work with in Coq.

3 RA and CMRA constructions

3.1 Product

Given a family $(M_i)_{i \in I}$ of CMRAs (I finite), we construct a CMRA for the product $\prod_{i \in I} M_i$ by lifting everything pointwise.

Frame-preserving updates on the M_i lift to the product:

$$\textsc{prod-update}\quad \frac{a \rightsquigarrow_{M_i} B}{f[i \mapsto a] \rightsquigarrow \{f[i \mapsto b] \mid b \in B\}}$$

3.2 Sum

The *sum CMRA* $M_1 +_\bot M_2$ for any CMRAs M_1 and M_2 is defined as (again, we use a datatype-like notation):

$$\begin{aligned}
M_1 +_\bot M_2 &\triangleq \mathsf{inl}(a_1 : M_1) \mid \mathsf{inr}(a_2 : M_2) \mid \bot \\
\mathcal{V}_n &\triangleq \{\mathsf{inl}(a_1) | a_1 \in \mathcal{V}'_n\} \cup \{\mathsf{inr}(a_2) | a_2 \in \mathcal{V}''_n\} \\
\mathsf{inl}(a_1) \cdot \mathsf{inl}(b_1) &\triangleq \mathsf{inl}(a_1 \cdot b_1) \\
|\mathsf{inl}(a_1)| &\triangleq \begin{cases} \top & \text{if } |a_1| = \top \\ \mathsf{inl}(|a_1|) & \text{otherwise} \end{cases}
\end{aligned}$$

The composition and core for inr are defined symmetrically. The remaining cases of the composition and core are all $\bot$. Above, $\mathcal{V}'$ refers to the validity of M_1, and $\mathcal{V}''$ to the validity of M_2.

We obtain the following frame-preserving updates, as well as their symmetric counterparts:

$$\textsc{sum-update}\quad \frac{a \rightsquigarrow_{M_1} B}{\mathsf{inl}(a) \rightsquigarrow \{\mathsf{inl}(b) \mid b \in B\}} \qquad \textsc{sum-swap}\quad \frac{\forall a_\mathrm{f}, n.\, a \cdot a_\mathrm{f} \notin \mathcal{V}'_n \qquad b \in \mathcal{V}''}{\mathsf{inl}(a) \rightsquigarrow \mathsf{inr}(b)}$$

Crucially, the second rule allows us to *swap* the “side” of the sum that the CMRA is on if $\mathcal{V}$ has *no possible frame*.

3.3 Finite partial function

Given some infinite countable K and some CMRA M, the set of finite partial functions $K \xrightarrow{\mathrm{fin}} M$ is equipped with a COFE and CMRA structure by lifting everything pointwise.

We obtain the following frame-preserving updates:

$$\textsc{fpfn-alloc-strong}\quad \frac{G \text{ infinite} \qquad a \in \mathcal{V}}{\emptyset \rightsquigarrow \{[\gamma \mapsto a] \mid \gamma \in G\}} \qquad \textsc{fpfn-alloc}\quad \frac{a \in \mathcal{V}}{\emptyset \rightsquigarrow \{[\gamma \mapsto a] \mid \gamma \in K\}} \qquad \textsc{fpfn-update}\quad \frac{a \rightsquigarrow_M B}{f[i \mapsto a] \rightsquigarrow \{f[i \mapsto b] \mid b \in B\}}$$

Above, $\mathcal{V}$ refers to the validity of M.

$K \xrightarrow{\mathrm{fin}} (-)$ is a locally non-expansive functor from $\mathcal{CMRA}$ to $\mathcal{CMRA}$.

3.4 Agreement

Given some COFE T, we define $\textsc{Ag}(T)$ as follows:

$$\begin{aligned}
\textsc{Ag}(T) &\triangleq \{(c, V) \in (\mathbb{N} \to T) \times \mathit{SProp}\} / \sim \\
&\quad \text{where } a \sim b \triangleq a.V = b.V \wedge \forall n.\, n \in a.V \Rightarrow a.c(n) \stackrel{n}{=} b.c(n) \\
a \stackrel{n}{=} b &\triangleq (\forall m \leq n.\, m \in a.V \Leftrightarrow m \in b.V) \wedge (\forall m \leq n.\, m \in a.V \Rightarrow a.c(m) \stackrel{m}{=} b.c(m)) \\
\mathcal{V}_n &\triangleq \left\{ a \in \textsc{Ag}(T) \,\middle|\, n \in a.V \wedge \forall m \leq n.\, a.c(n) \stackrel{m}{=} a.c(m) \right\} \\
|a| &\triangleq a \\
a \cdot b &\triangleq \left(a.c, \left\{ n \,\middle|\, n \in a.V \wedge n \in b.V \wedge a \stackrel{n}{=} b \right\} \right)
\end{aligned}$$

$\textsc{Ag}(-)$ is a locally non-expansive functor from $\mathcal{COFE}$ to $\mathcal{CMRA}$.

You can think of the c as a *chain* of elements of T that has to converge only for $n \in V$ steps. The reason we store a chain, rather than a single element, is that $\textsc{Ag}(T)$ needs to be a COFE itself, so we need to be able to give a limit for every chain of $\textsc{Ag}(T)$. However, given such a chain, we cannot constructively define its limit: Clearly, the V of the limit is the limit of the V of the chain. But what to pick for the actual data, for the element of T? Only if $V = \mathbb{N}$ we have a chain of T that we can take a limit of; if the V is smaller, the chain "cancels", *i.e.,* stops converging as we reach indices $n \notin V$. To mitigate this, we apply the usual construction to close a set; we go from elements of T to chains of T.

We define an injection ag into $\textsc{Ag}(T)$ as follows:

$$\mathsf{ag}(x) \triangleq \left\{ \mathrm{c} \triangleq \lambda\_.\, x, \mathrm{V} \triangleq \mathbb{N} \right\}$$

There are no interesting frame-preserving updates for $\textsc{Ag}(T)$, but we can show the following:

$$\frac{\textsc{ag-val}}{\mathsf{ag}(x) \in \mathcal{V}_n} \qquad \frac{\textsc{ag-dup}}{\mathsf{ag}(x) = \mathsf{ag}(x) \cdot \mathsf{ag}(x)} \qquad \frac{\textsc{ag-agree}}{\mathsf{ag}(x) \cdot \mathsf{ag}(y) \in \mathcal{V}_n \Rightarrow x \stackrel{n}{=} y}$$

3.5 Exclusive CMRA

Given a COFE T, we define a CMRA $\textsc{Ex}(T)$ such that at most one $x \in T$ can be owned:

$$\begin{aligned}
\textsc{Ex}(T) &\triangleq \mathsf{ex}(T) + \bot \\
\mathcal{V}_n &\triangleq \{a \in \textsc{Ex}(T) \mid a \neq \bot\}
\end{aligned}$$

All cases of composition go to $\bot$.

$$|\mathsf{ex}(x)| \triangleq \top \qquad\qquad |\bot| \triangleq \bot$$

Remember that $\top$ is the "dummy" element in $M^?$ indicating (in this case) that $\mathsf{ex}(x)$ has no core.

The step-indexed equivalence is inductively defined as follows:

$$\frac{x \stackrel{n}{=} y}{\mathsf{ex}(x) \stackrel{n}{=} \mathsf{ex}(y)} \qquad\qquad \bot \stackrel{n}{=} \bot$$

$\textsc{Ex}(-)$ is a locally non-expansive functor from $\mathcal{COFE}$ to $\mathcal{CMRA}$.

We obtain the following frame-preserving update:

$$\frac{\textsc{ex-update}}{\mathsf{ex}(x) \rightsquigarrow \mathsf{ex}(y)}$$

3.6 STS with tokens

Given a state-transition system (STS, *i.e.,* a directed graph) $(\mathcal{S}, \rightarrow \subseteq \mathcal{S} \times \mathcal{S})$, a set of tokens $\mathcal{T}$, and a labeling $\mathcal{L} : \mathcal{S} \rightarrow \wp(\mathcal{T})$ of *protocol-owned* tokens for each state, we construct an RA modeling an authoritative current state and permitting transitions given a *bound* on the current state and a set of *locally-owned* tokens.

The construction follows the idea of STSs as described in CaReSL [4]. We first lift the transition relation to $\mathcal{S} \times \wp(\mathcal{T})$ (implementing a *law of token conservation*) and define a stepping relation for the *frame* of a given token set:

$$(s, T) \rightarrow (s', T') \triangleq s \rightarrow s' \wedge \mathcal{L}(s) \uplus T = \mathcal{L}(s') \uplus T'$$

$$s \xrightarrow{T} s' \triangleq \exists T_1, T_2.\, T_1 \mathrel{\#} \mathcal{L}(s) \cup T \wedge (s, T_1) \rightarrow (s', T_2)$$

We further define *closed* sets of states (given a particular set of tokens) as well as the *closure* of a set:

$$\mathsf{closed}(S, T) \triangleq \forall s \in S.\, \mathcal{L}(s) \mathrel{\#} T \wedge \left(\forall s'.\, s \xrightarrow{T} s' \Rightarrow s' \in S\right)$$

$$\uparrow(S, T) \triangleq \left\{ s' \in \mathcal{S} \;\middle|\; \exists s \in S.\, s \xrightarrow{T}{}^{*} s' \right\}$$

The STS RA is defined as follows

$$\begin{aligned} M \triangleq\ & \{\mathsf{auth}((s, T) \in \mathcal{S} \times \wp(\mathcal{T})) \mid \mathcal{L}(s) \mathrel{\#} T\} + \\ & \{\mathsf{frag}((S, T) \in \wp(\mathcal{S}) \times \wp(\mathcal{T})) \mid \mathsf{closed}(S, T) \wedge S \neq \emptyset\} + \bot \end{aligned}$$

$$\mathsf{frag}(S_1, T_1) \cdot \mathsf{frag}(S_2, T_2) \triangleq \mathsf{frag}(S_1 \cap S_2, T_1 \cup T_2) \qquad \text{if } T_1 \mathrel{\#} T_2 \text{ and } S_1 \cap S_2 \neq \emptyset$$

$$\mathsf{frag}(S, T) \cdot \mathsf{auth}(s, T') \triangleq \mathsf{auth}(s, T') \cdot \mathsf{frag}(S, T) \triangleq \mathsf{auth}(s, T \cup T') \qquad \text{if } T \mathrel{\#} T' \text{ and } s \in S$$

$$|\mathsf{frag}(S, T)| \triangleq \mathsf{frag}(\uparrow(S, \emptyset), \emptyset)$$

$$|\mathsf{auth}(s, T)| \triangleq \mathsf{frag}(\uparrow(\{s\}, \emptyset), \emptyset)$$

The remaining cases are all $\bot$.

We will need the following frame-preserving update:

STS-STEP
$$\frac{(s, T) \rightarrow^{*} (s', T')}{\mathsf{auth}(s, T) \rightsquigarrow \mathsf{auth}(s', T')}$$

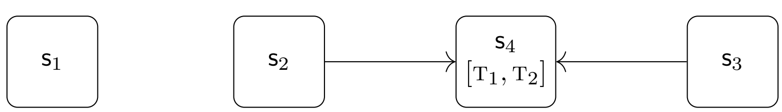
STS-WEAKEN
$$\frac{\mathsf{closed}(S_2, T_2) \qquad S_1 \subseteq S_2 \qquad T_2 \subseteq T_1}{\mathsf{frag}(S_1, T_1) \rightsquigarrow \mathsf{frag}(S_2, T_2)}$$

The core is not a homomorphism. The core of the STS construction is only satisfying the RA axioms because we are *not* demanding the core to be a homomorphism—all we demand is for the core to be monotone with respect the RA-INCL.

In other words, the following does *not* hold for the STS core as defined above:

$$|a| \cdot |b| = |a \cdot b|$$

To see why, consider the following STS:



Now consider the following two elements of the STS RA:

$$a \triangleq \mathsf{frag}(\{\mathsf{s}_1, \mathsf{s}_2\}, \{\mathrm{T}_1\}) \qquad b \triangleq \mathsf{frag}(\{\mathsf{s}_1, \mathsf{s}_3\}, \{\mathrm{T}_2\})$$

We have:

$$a \cdot b = \mathsf{frag}(\{\mathsf{s}_1\}, \{\mathrm{T}_1, \mathrm{T}_2\}) \qquad |a| = \mathsf{frag}(\{\mathsf{s}_1, \mathsf{s}_2, \mathsf{s}_4\}, \emptyset) \qquad |b| = \mathsf{frag}(\{\mathsf{s}_1, \mathsf{s}_3, \mathsf{s}_4\}, \emptyset)$$

$$|a| \cdot |b| = \mathsf{frag}(\{\mathsf{s}_1, \mathsf{s}_4\}, \emptyset) \neq |a \cdot b| = \mathsf{frag}(\{\mathsf{s}_1\}, \emptyset)$$

4 Language

A *language* Λ consists of a set *Expr* of *expressions* (metavariable e), a set *Val* of *values* (metavariable v), and a set *State* of *states* (metvariable σ) such that

- There exist functions val2expr : $\mathit{Val} \to \mathit{Expr}$ and expr2val : $\mathit{Expr} \rightharpoonup \mathit{val}$ (notice the latter is partial), such that

$$\forall e, v.\, \mathrm{expr2val}(e) = v \Rightarrow \mathrm{val2expr}(v) = e \qquad\qquad \forall v.\, \mathrm{expr2val}(\mathrm{val2expr}(v)) = v$$

- There exists a *primitive reduction relation*

$$(-, - \to -, -, -) \subseteq \mathit{Expr} \times \mathit{State} \times \mathit{Expr} \times \mathit{State} \times (\mathit{Expr} \uplus \{\bot\})$$

  We will write $e_1, \sigma_1 \to e_2, \sigma_2$ for $e_1, \sigma_1 \to e_2, \sigma_2, \bot$.
  A reduction $e_1, \sigma_1 \to e_2, \sigma_2, e_\mathrm{f}$ indicates that, when e_1 reduces to e_2, a *new thread* e_f is forked off.

- All values are stuck:

$$e, \_ \to \_, \_, \_ \Rightarrow \mathrm{expr2val}(e) = \bot$$

Definition 16. *An expression e and state σ are* reducible *(written* $\mathrm{red}(e, \sigma)$*) if*

$$\exists e_2, \sigma_2, e_\mathrm{f}.\, e, \sigma \to e_2, \sigma_2, e_\mathrm{f}$$

Definition 17. *An expression e is said to be* atomic *if it reduces in one step to a value:*

$$\forall \sigma_1, e_2, \sigma_2, e_\mathrm{f}.\, e, \sigma_1 \to e_2, \sigma_2, e_\mathrm{f} \Rightarrow \exists v_2.\, \mathrm{expr2val}(e_2) = v_2$$

Definition 18 (Context)**.** *A function $K : \mathit{Expr} \to \mathit{Expr}$ is a* context *if the following conditions are satisfied:*

1. *K does not turn non-values into values:*
   $\forall e.\, \mathrm{expr2val}(e) = \bot \Rightarrow \mathrm{expr2val}(K(e)) = \bot$
2. *One can perform reductions below K:*
   $\forall e_1, \sigma_1, e_2, \sigma_2, e_\mathrm{f}.\, e_1, \sigma_1 \to e_2, \sigma_2, e_\mathrm{f} \Rightarrow K(e_1), \sigma_1 \to K(e_2), \sigma_2, e_\mathrm{f}$
3. *Reductions stay below K until there is a value in the hole:*
   $\forall e'_1, \sigma_1, e_2, \sigma_2, e_\mathrm{f}.\, \mathrm{expr2val}(e'_1) = \bot \wedge K(e'_1), \sigma_1 \to e_2, \sigma_2, e_\mathrm{f} \Rightarrow \exists e'_2.\, e_2 = K(e'_2) \wedge e'_1, \sigma_1 \to e'_2, \sigma_2, e_\mathrm{f}$

4.1 Concurrent language

For any language Λ, we define the corresponding thread-pool semantics.

Machine syntax

$$T \in \mathit{ThreadPool} \triangleq \bigcup_n \mathit{Expr}^n$$

Machine reduction $\boxed{T; \sigma \to T'; \sigma'}$

$$\frac{e_1, \sigma_1 \to e_2, \sigma_2, e_\mathrm{f} \qquad e_\mathrm{f} \neq \bot}{T \mathbin{+\!\!+} [e_1] \mathbin{+\!\!+} T'; \sigma_1 \to T \mathbin{+\!\!+} [e_2] \mathbin{+\!\!+} T' \mathbin{+\!\!+} [e_\mathrm{f}]; \sigma_2} \qquad \frac{e_1, \sigma_1 \to e_2, \sigma_2}{T \mathbin{+\!\!+} [e_1] \mathbin{+\!\!+} T'; \sigma_1 \to T \mathbin{+\!\!+} [e_2] \mathbin{+\!\!+} T'; \sigma_2}$$

5 Logic

To instantiate Iris, you need to define the following parameters:

- A language Λ, and
- a locally contractive bifunctor $\Sigma : \mathcal{COFE} \to \mathcal{CMRA}$ defining the ghost state, such that for all COFEs A, the CMRA $\Sigma(A)$ has a unit. (By Lemma 1, this means that the core of $\Sigma(A)$ is a total function.)

As usual for higher-order logics, you can furthermore pick a *signature* $\mathcal{S} = (\mathcal{T}, \mathcal{F}, \mathcal{A})$ to add more types, symbols and axioms to the language. You have to make sure that $\mathcal{T}$ includes the base types:

$$\mathcal{T} \supseteq \{\mathsf{Val}, \mathsf{Expr}, \mathsf{State}, \mathsf{M}, \mathsf{InvName}, \mathsf{InvMask}, \mathsf{Prop}\}$$

Elements of $\mathcal{T}$ are ranged over by T.

Each function symbol in $\mathcal{F}$ has an associated *arity* comprising a natural number n and an ordered list of $n+1$ types τ (the grammar of τ is defined below, and depends only on $\mathcal{T}$). We write

$$F : \tau_1, \ldots, \tau_n \to \tau_{n+1} \in \mathcal{F}$$

to express that F is a function symbol with the indicated arity.

Furthermore, $\mathcal{A}$ is a set of *axioms*, that is, terms t of type Prop. Again, the grammar of terms and their typing rules are defined below, and depends only on $\mathcal{T}$ and $\mathcal{F}$, not on $\mathcal{A}$. Elements of $\mathcal{A}$ are ranged over by A.

5.1 Grammar

Syntax. Iris syntax is built up from a signature $\mathcal{S}$ and a countably infinite set *Var* of variables (ranged over by metavariables x, y, z):

$$\begin{aligned}
\tau ::={}& T \mid 1 \mid \tau \times \tau \mid \tau \to \tau \\
t, P, \varphi ::={}& x \mid F(t_1, \ldots, t_n) \mid () \mid (t, t) \mid \pi_i\, t \mid \lambda x : \tau.\, t \mid t(t) \mid \varepsilon \mid |t| \mid t \cdot t \mid \\
& \mathsf{False} \mid \mathsf{True} \mid t =_\tau t \mid P \Rightarrow P \mid P \land P \mid P \lor P \mid P * P \mid P -\!\!* P \mid \\
& \mu x : \tau.\, t \mid \exists x : \tau.\, P \mid \forall x : \tau.\, P \mid \\
& \boxed{P}^{t} \mid \boxed{t} \mid \mathcal{V}(t) \mid \mathsf{Phy}(t) \mid \Box P \mid \triangleright P \mid {}^{t}\!\Rrightarrow^{t} P \mid \mathsf{wp}_{t}\, t\, \{x.\, t\}
\end{aligned}$$

Recursive predicates must be *guarded*: in $\mu x.\, t$, the variable x can only appear under the later $\triangleright$ modality.

Note that $\Box$ and $\triangleright$ bind more tightly than $*$, $-\!\!*$, $\land$, $\lor$, and $\Rightarrow$. We will write $\Rrightarrow_t P$ for ${}^{t}\!\Rrightarrow^{t} P$. If we omit the mask, then it is $\top$ for weakest precondition $\mathsf{wp}\, e\, \{x.\, P\}$ and $\emptyset$ for primitive view shifts $\Rrightarrow P$.

Some propositions are *timeless*, which intuitively means that step-indexing does not affect them. This is a *meta-level* assertion about propositions, defined as follows:

$$\Gamma \vdash \mathsf{timeless}(P) \triangleq \Gamma \mid \triangleright P \vdash P \lor \triangleright \mathsf{False}$$

Metavariable conventions. We introduce additional metavariables ranging over terms and generally let the choice of metavariable indicate the term's type:

metavariable	type
t, u	arbitrary
v, w	Val
e	Expr
σ	State

metavariable	type
ι	InvName
$\mathcal{E}$	InvMask
a, b	M
P, Q, R	Prop
φ, ψ, ζ	$\tau \to \mathsf{Prop}$ (when τ is clear from context)

Variable conventions. We assume that, if a term occurs multiple times in a rule, its free variables are exactly those binders which are available at every occurrence.

5.2 Types

Iris terms are simply-typed. The judgment $\Gamma \vdash t : \tau$ expresses that, in variable context Γ, the term t has type τ.

A variable context, $\Gamma = x_1 : \tau_1, \ldots, x_n : \tau_n$, declares a list of variables and their types. In writing $\Gamma, x : \tau$, we presuppose that x is not already declared in Γ.

Well-typed terms $\boxed{\Gamma \vdash_{\mathcal{S}} t : \tau}$

$$x : \tau \vdash x : \tau \qquad \frac{\Gamma \vdash t : \tau}{\Gamma, x : \tau' \vdash t : \tau} \qquad \frac{\Gamma, x : \tau', y : \tau' \vdash t : \tau}{\Gamma, x : \tau' \vdash t[x/y] : \tau} \qquad \frac{\Gamma_1, x : \tau', y : \tau'', \Gamma_2 \vdash t : \tau}{\Gamma_1, x : \tau'', y : \tau', \Gamma_2 \vdash t[y/x, x/y] : \tau}$$

$$\frac{\Gamma \vdash t_1 : \tau_1 \qquad \cdots \qquad \Gamma \vdash t_n : \tau_n \qquad F : \tau_1, \ldots, \tau_n \to \tau_{n+1} \in \mathcal{F}}{\Gamma \vdash F(t_1, \ldots, t_n) : \tau_{n+1}} \qquad \Gamma \vdash () : 1$$

$$\frac{\Gamma \vdash t : \tau_1 \qquad \Gamma \vdash u : \tau_2}{\Gamma \vdash (t, u) : \tau_1 \times \tau_2} \qquad \frac{\Gamma \vdash t : \tau_1 \times \tau_2 \qquad i \in \{1, 2\}}{\Gamma \vdash \pi_i\, t : \tau_i} \qquad \frac{\Gamma, x : \tau \vdash t : \tau'}{\Gamma \vdash \lambda x.\, t : \tau \to \tau'}$$

$$\frac{\Gamma \vdash t : \tau \to \tau' \qquad u : \tau}{\Gamma \vdash t(u) : \tau'} \qquad \Gamma \vdash \varepsilon : \mathsf{M} \qquad \frac{\Gamma \vdash a : \mathsf{M}}{\Gamma \vdash |a| : \mathsf{M}} \qquad \frac{\Gamma \vdash a : \mathsf{M} \qquad \Gamma \vdash b : \mathsf{M}}{\Gamma \vdash a \cdot b : \mathsf{M}}$$

$$\Gamma \vdash \mathsf{False} : \mathsf{Prop} \qquad \Gamma \vdash \mathsf{True} : \mathsf{Prop} \qquad \frac{\Gamma \vdash t : \tau \qquad \Gamma \vdash u : \tau}{\Gamma \vdash t =_\tau u : \mathsf{Prop}} \qquad \frac{\Gamma \vdash P : \mathsf{Prop} \qquad \Gamma \vdash Q : \mathsf{Prop}}{\Gamma \vdash P \Rightarrow Q : \mathsf{Prop}}$$

$$\frac{\Gamma \vdash P : \mathsf{Prop} \qquad \Gamma \vdash Q : \mathsf{Prop}}{\Gamma \vdash P \wedge Q : \mathsf{Prop}} \qquad \frac{\Gamma \vdash P : \mathsf{Prop} \qquad \Gamma \vdash Q : \mathsf{Prop}}{\Gamma \vdash P \vee Q : \mathsf{Prop}} \qquad \frac{\Gamma \vdash P : \mathsf{Prop} \qquad \Gamma \vdash Q : \mathsf{Prop}}{\Gamma \vdash P * Q : \mathsf{Prop}}$$

$$\frac{\Gamma \vdash P : \mathsf{Prop} \qquad \Gamma \vdash Q : \mathsf{Prop}}{\Gamma \vdash P \mathrel{-\!\!*} Q : \mathsf{Prop}} \qquad \frac{\Gamma, x : \tau \vdash t : \tau \qquad x \text{ is guarded in } t}{\Gamma \vdash \mu x : \tau.\, t : \tau} \qquad \frac{\Gamma, x : \tau \vdash P : \mathsf{Prop}}{\Gamma \vdash \exists x : \tau.\, P : \mathsf{Prop}}$$

$$\frac{\Gamma, x : \tau \vdash P : \mathsf{Prop}}{\Gamma \vdash \forall x : \tau.\, P : \mathsf{Prop}} \qquad \frac{\Gamma \vdash P : \mathsf{Prop} \qquad \Gamma \vdash \iota : \mathsf{InvName}}{\Gamma \vdash \boxed{P}^{\iota} : \mathsf{Prop}} \qquad \frac{\Gamma \vdash a : \mathsf{M}}{\Gamma \vdash \boxed{a} : \mathsf{Prop}}$$

$$\frac{\Gamma \vdash a : \tau \qquad \tau \text{ is a CMRA}}{\Gamma \vdash \mathcal{V}(a) : \mathsf{Prop}} \qquad \frac{\Gamma \vdash \sigma : \mathsf{State}}{\Gamma \vdash \mathsf{Phy}(\sigma) : \mathsf{Prop}} \qquad \frac{\Gamma \vdash P : \mathsf{Prop}}{\Gamma \vdash \Box P : \mathsf{Prop}} \qquad \frac{\Gamma \vdash P : \mathsf{Prop}}{\Gamma \vdash \triangleright P : \mathsf{Prop}}$$

$$\frac{\Gamma \vdash P : \mathsf{Prop} \qquad \Gamma \vdash \mathcal{E} : \mathsf{InvMask} \qquad \Gamma \vdash \mathcal{E}' : \mathsf{InvMask}}{\Gamma \vdash {}^{\mathcal{E}}{\Rrightarrow}^{\mathcal{E}'} P : \mathsf{Prop}}$$

$$\frac{\Gamma \vdash e : \mathsf{Expr} \qquad \Gamma, x : \mathsf{Val} \vdash t : \mathsf{Prop} \qquad \Gamma \vdash \mathcal{E} : \mathsf{InvMask}}{\Gamma \vdash \mathsf{wp}_{\mathcal{E}}\; e\, \{x.\, t\} : \mathsf{Prop}}$$

5.3 Proof rules

The judgment $\Gamma \mid \Theta \vdash P$ says that with free variables Γ, proposition P holds whenever all assumptions Θ hold. We implicitly assume that an arbitrary variable context, Γ, is added to every constituent of the rules. Furthermore, an arbitrary *boxed* assertion context $\Box\Theta$ may be added to every constituent. Axioms $\Gamma \mid P \dashv\vdash Q$ indicate that both $\Gamma \mid P \vdash Q$ and $\Gamma \mid Q \vdash P$ can be derived.

$$\boxed{\Gamma \mid \Theta \vdash P}$$

Laws of intuitionistic higher-order logic with equality. This is entirely standard.

$$\textsc{Asm}\ \frac{P \in \Theta}{\Theta \vdash P} \qquad \textsc{Eq}\ \frac{\Theta \vdash P \qquad \Theta \vdash t =_\tau t'}{\Theta \vdash P[t'/t]} \qquad \textsc{Refl}\ \Theta \vdash t =_\tau t \qquad \bot\mathrm{E}\ \frac{\Theta \vdash \mathsf{False}}{\Theta \vdash P} \qquad \top\mathrm{I}\ \Theta \vdash \mathsf{True} \qquad \wedge\mathrm{I}\ \frac{\Theta \vdash P \qquad \Theta \vdash Q}{\Theta \vdash P \wedge Q}$$

$$\wedge\mathrm{EL}\ \frac{\Theta \vdash P \wedge Q}{\Theta \vdash P} \qquad \wedge\mathrm{ER}\ \frac{\Theta \vdash P \wedge Q}{\Theta \vdash Q} \qquad \vee\mathrm{IL}\ \frac{\Theta \vdash P}{\Theta \vdash P \vee Q} \qquad \vee\mathrm{IR}\ \frac{\Theta \vdash Q}{\Theta \vdash P \vee Q} \qquad \vee\mathrm{E}\ \frac{\Theta \vdash P \vee Q \qquad \Theta, P \vdash R \qquad \Theta, Q \vdash R}{\Theta \vdash R}$$

$$\Rightarrow\mathrm{I}\ \frac{\Theta, P \vdash Q}{\Theta \vdash P \Rightarrow Q} \qquad \Rightarrow\mathrm{E}\ \frac{\Theta \vdash P \Rightarrow Q \qquad \Theta \vdash P}{\Theta \vdash Q} \qquad \forall\mathrm{I}\ \frac{\Gamma, x : \tau \mid \Theta \vdash P}{\Gamma \mid \Theta \vdash \forall x : \tau.\ P} \qquad \forall\mathrm{E}\ \frac{\Gamma \mid \Theta \vdash \forall x : \tau.\ P \qquad \Gamma \vdash t : \tau}{\Gamma \mid \Theta \vdash P[t/x]}$$

$$\exists\mathrm{I}\ \frac{\Gamma \mid \Theta \vdash P[t/x] \qquad \Gamma \vdash t : \tau}{\Gamma \mid \Theta \vdash \exists x : \tau. P} \qquad \exists\mathrm{E}\ \frac{\Gamma \mid \Theta \vdash \exists x : \tau.\ P \qquad \Gamma, x : \tau \mid \Theta, P \vdash Q}{\Gamma \mid \Theta \vdash Q}$$

Furthermore, we have the usual η and β laws for projections, λ and μ.

Laws of (affine) bunched implications.

$$\begin{aligned} \mathsf{True} * P &\dashv\vdash P \\ P * Q &\dashv\vdash Q * P \\ (P * Q) * R &\dashv\vdash P * (Q * R) \end{aligned} \qquad *\text{-MONO}\ \frac{P_1 \vdash Q_1 \qquad P_2 \vdash Q_2}{P_1 * P_2 \vdash Q_1 * Q_2} \qquad \mathrel{-\!\!*}\ \text{I-E}\ \frac{P * Q \vdash R}{P \vdash Q \mathrel{-\!\!*} R}$$

Laws for ghosts and physical resources.

$$\begin{aligned} \boxed{a} * \boxed{b} &\dashv\vdash \boxed{a \cdot b} \\ \boxed{a} &\vdash \mathcal{V}(a) \\ \mathsf{True} &\vdash \boxed{\varepsilon} \end{aligned} \qquad \mathsf{Phy}(\sigma) * \mathsf{Phy}(\sigma') \vdash \mathsf{False}$$

Laws for the later modality.

$$\triangleright\text{-MONO}\ \frac{\Theta \vdash P}{\Theta \vdash \triangleright P} \qquad \textsc{Löb}\ (\triangleright P \Rightarrow P) \vdash P \qquad \triangleright\text{-}\exists\ \frac{\tau \text{ is inhabited}}{\triangleright \exists x : \tau.\ P \vdash \exists x : \tau.\ \triangleright P}$$

$$\begin{aligned} \triangleright(P \wedge Q) &\dashv\vdash \triangleright P \wedge \triangleright Q \\ \triangleright(P \vee Q) &\dashv\vdash \triangleright P \vee \triangleright Q \end{aligned} \qquad \begin{aligned} \triangleright \forall x.\ P &\dashv\vdash \forall x.\ \triangleright P \\ \exists x.\ \triangleright P &\vdash \triangleright \exists x.\ P \\ \triangleright(P * Q) &\dashv\vdash \triangleright P * \triangleright Q \end{aligned}$$

A type τ being *inhabited* means that $\vdash t : \tau$ is derivable for some t.

$$\frac{t \text{ or } t' \text{ is a discrete COFE element}}{\mathsf{timeless}(t =_\tau t')} \qquad \frac{a \text{ is a discrete COFE element}}{\mathsf{timeless}(\bbox[border:1px dashed]{a})}$$

$$\frac{a \text{ is an element of a discrete CMRA}}{\mathsf{timeless}(\mathcal{V}(a))} \qquad \mathsf{timeless}(\mathsf{Phy}(\sigma)) \qquad \frac{\Gamma \vdash \mathsf{timeless}(Q)}{\Gamma \vdash \mathsf{timeless}(P \Rightarrow Q)}$$

$$\frac{\Gamma \vdash \mathsf{timeless}(Q)}{\Gamma \vdash \mathsf{timeless}(P \mathrel{-\!\!*} Q)} \qquad \frac{\Gamma, x : \tau \vdash \mathsf{timeless}(P)}{\Gamma \vdash \mathsf{timeless}(\forall x : \tau.\, P)} \qquad \frac{\Gamma, x : \tau \vdash \mathsf{timeless}(P)}{\Gamma \vdash \mathsf{timeless}(\exists x : \tau.\, P)}$$

Laws for the always modality.

$\Box$I
$$\frac{\Box\Theta \vdash P}{\Box\Theta \vdash \Box P}$$

$\Box$E
$$\Box P \vdash P$$

$$\Box(P \land Q) \;\vdash\; \Box(P * Q)$$
$$\Box P \land Q \;\vdash\; \Box P * Q$$
$$\Box \triangleright P \dashv\vdash \triangleright \Box P$$

$$\Box(P \land Q) \dashv\vdash \Box P \land \Box Q$$
$$\Box(P \lor Q) \dashv\vdash \Box P \lor \Box Q$$
$$\Box \forall x.\, P \dashv\vdash \forall x.\, \Box P$$
$$\Box \exists x.\, P \dashv\vdash \exists x.\, \Box P$$

$$t =_\tau t' \vdash \Box t =_\tau t' \qquad \boxed{P}^{\iota} \vdash \Box \boxed{P}^{\iota} \qquad \bbox[border:1px dashed]{a} \vdash \Box \bbox[border:1px dashed]{a} \qquad \mathcal{V}(a) \vdash \Box \mathcal{V}(a)$$

Laws of primitive view shifts.

PVS-INTRO
$$P \vdash \Rrightarrow_{\mathcal{E}} P$$

PVS-MONO
$$\frac{P \vdash Q}{{}^{\mathcal{E}_1}\!\Rrightarrow^{\mathcal{E}_2} P \vdash {}^{\mathcal{E}_1}\!\Rrightarrow^{\mathcal{E}_2} Q}$$

PVS-TIMELESS
$$\frac{\mathsf{timeless}(P)}{\triangleright P \vdash \Rrightarrow_{\mathcal{E}} P}$$

PVS-TRANS
$$\frac{\mathcal{E}_2 \subseteq \mathcal{E}_1 \cup \mathcal{E}_3}{{}^{\mathcal{E}_1}\!\Rrightarrow^{\mathcal{E}_2}\, {}^{\mathcal{E}_2}\!\Rrightarrow^{\mathcal{E}_3} P \vdash {}^{\mathcal{E}_1}\!\Rrightarrow^{\mathcal{E}_3} P}$$

PVS-MASK-FRAME
$${}^{\mathcal{E}_1}\!\Rrightarrow^{\mathcal{E}_2} P \vdash {}^{\mathcal{E}_1 \uplus \mathcal{E}_\mathrm{f}}\!\Rrightarrow^{\mathcal{E}_2 \uplus \mathcal{E}_\mathrm{f}} P$$

PVS-FRAME
$$Q * {}^{\mathcal{E}_1}\!\Rrightarrow^{\mathcal{E}_2} P \vdash {}^{\mathcal{E}_1}\!\Rrightarrow^{\mathcal{E}_2} Q * P$$

PVS-ALLOCI
$$\frac{\mathcal{E} \text{ is infinite}}{\triangleright P \vdash \Rrightarrow_{\mathcal{E}} \exists \iota \in \mathcal{E}.\, \boxed{P}^{\iota}}$$

PVS-OPENI
$$\boxed{P}^{\iota} \vdash {}^{\{\iota\}}\!\Rrightarrow^{\emptyset} \triangleright P$$

PVS-CLOSEI
$$\boxed{P}^{\iota} \land \triangleright P \vdash {}^{\emptyset}\!\Rrightarrow^{\{\iota\}} \mathsf{True}$$

PVS-UPDATE
$$\frac{a \rightsquigarrow B}{\bbox[border:1px dashed]{a} \vdash \Rrightarrow_{\mathcal{E}} \exists b \in B.\, \bbox[border:1px dashed]{b}}$$

Laws of weakest preconditions.

WP-VALUE
$$P[v/x] \vdash \mathsf{wp}_{\mathcal{E}}\; v \;\{x.\, P\}$$

WP-MONO
$$\frac{\mathcal{E}_1 \subseteq \mathcal{E}_2 \qquad x : \mathsf{val} \mid P \vdash Q}{\mathsf{wp}_{\mathcal{E}_1}\; e \;\{x.\, P\} \vdash \mathsf{wp}_{\mathcal{E}_2}\; e \;\{x.\, Q\}}$$

PVS-WP
$$\Rrightarrow_{\mathcal{E}} \mathsf{wp}_{\mathcal{E}}\; e \;\{x.\, P\} \vdash \mathsf{wp}_{\mathcal{E}}\; e \;\{x.\, P\}$$

WP-PVS
$$\mathsf{wp}_{\mathcal{E}}\; e \;\{x.\, \Rrightarrow_{\mathcal{E}} P\} \vdash \mathsf{wp}_{\mathcal{E}}\; e \;\{x.\, P\}$$

WP-ATOMIC
$$\frac{\mathcal{E}_2 \subseteq \mathcal{E}_1 \qquad \mathsf{atomic}(\mathrm{e})}{{}^{\mathcal{E}_1}\!\Rrightarrow^{\mathcal{E}_2} \mathsf{wp}_{\mathcal{E}_2}\; e \;\{x.\, {}^{\mathcal{E}_2}\!\Rrightarrow^{\mathcal{E}_1} P\} \vdash \mathsf{wp}_{\mathcal{E}_1}\; e \;\{x.\, P\}}$$

WP-FRAME
$$Q * \mathsf{wp}_{\mathcal{E}}\; e \;\{x.\, P\} \vdash \mathsf{wp}_{\mathcal{E}}\; e \;\{x.\, Q * P\}$$

WP-FRAME-STEP
$$\frac{\mathrm{expr2val}(e) = \bot \qquad \mathcal{E}_2 \subseteq \mathcal{E}_1}{\mathsf{wp}_{\mathcal{E}}\; e \;\{x.\, P\} * {}^{\mathcal{E}_1}\!\Rrightarrow^{\mathcal{E}_2} \triangleright\, {}^{\mathcal{E}_2}\!\Rrightarrow^{\mathcal{E}_1} Q \vdash \mathsf{wp}_{\mathcal{E} \uplus \mathcal{E}_1}\; e \;\{x.\, Q * P\}}$$

WP-BIND
$$\frac{K \text{ is a context}}{\mathsf{wp}_{\mathcal{E}}\; e \;\{x.\, \mathsf{wp}_{\mathcal{E}}\; K(\mathrm{val2expr}(x)) \;\{y.\, P\}\} \vdash \mathsf{wp}_{\mathcal{E}}\; K(e) \;\{y.\, P\}}$$

Lifting of operational semantics.

WP-LIFT-STEP

$$\frac{\mathcal{E}_2 \subseteq \mathcal{E}_1 \qquad \text{expr2val}(e_1) = \bot}{\begin{array}{l} {}^{\mathcal{E}_1}\!\!\Rrightarrow^{\mathcal{E}_2} \exists \sigma_1.\, \text{red}(e_1, \sigma_1) \land \triangleright \mathsf{Phy}(\sigma_1) \ast \\ \qquad \triangleright \forall e_2, \sigma_2, e_{\text{f}}.\ ((e_1, \sigma_1 \to e_2, \sigma_2, e_{\text{f}}) \land \mathsf{Phy}(\sigma_2)) \mathrel{-\!\!\ast} {}^{\mathcal{E}_2}\!\!\Rrightarrow^{\mathcal{E}_1} \mathsf{wp}_{\mathcal{E}_1}\, e_2\, \{x.\, P\} \ast \mathsf{wp}_{\top}\, e_{\text{f}}\, \{\_.\, \mathsf{True}\} \\ \vdash \mathsf{wp}_{\mathcal{E}_1}\, e_1\, \{x.\, P\} \end{array}}$$

WP-LIFT-PURE-STEP

$$\frac{\text{expr2val}(e_1) = \bot \qquad \forall \sigma_1.\, \text{red}(e_1, \sigma_1) \qquad \forall \sigma_1, e_2, \sigma_2, e_{\text{f}}.\, e_1, \sigma_1 \to e_2, \sigma_2, e_{\text{f}} \Rightarrow \sigma_1 = \sigma_2}{\triangleright \forall \sigma, e_2, e_{\text{f}}.\, (e_1, \sigma \to e_2, \sigma, e_{\text{f}}) \Rightarrow \mathsf{wp}_{\mathcal{E}_1}\, e_2\, \{x.\, P\} \ast \mathsf{wp}_{\top}\, e_{\text{f}}\, \{\_.\, \mathsf{True}\} \vdash \mathsf{wp}_{\mathcal{E}_1}\, e_1\, \{x.\, P\}}$$

Notice that primitive view shifts cover everything to their right, *i.e.,* $\Rrightarrow P \ast Q \triangleq \Rrightarrow (P \ast Q)$.

Here we define $\mathsf{wp}_{\mathcal{E}}\, e_{\text{f}}\, \{x.\, P\} \triangleq \mathsf{True}$ if $e_{\text{f}} = \bot$ (remember that our stepping relation can, but does not have to, define a forked-off expression).

5.4 Adequacy

The adequacy statement concerning functional correctness reads as follows:

$$\begin{array}{l} \forall \mathcal{E}, e, v, \varphi, \sigma, a, \sigma', T'. \\ (\forall n.\, a \in \mathcal{V}_n) \Rightarrow \\ (\mathsf{Phy}(\sigma) \ast \boxed{a} \vdash \mathsf{wp}_{\mathcal{E}}\, e\, \{x.\, \varphi(x)\}) \Rightarrow \\ \sigma; [e] \to^{*} \sigma'; [v] \mathbin{+\!\!+} T' \Rightarrow \\ \varphi(v) \end{array}$$

where φ is a *meta-level* predicate over values, *i.e.,* it can mention neither resources nor invariants.

Furthermore, the following adequacy statement shows that our weakest preconditions imply that the execution never gets *stuck*: Every expression in the thread pool either is a value, or can reduce further.

$$\begin{array}{l} \forall \mathcal{E}, e, \sigma, a, \sigma', T'. \\ (\forall n.\, a \in \mathcal{V}_n) \Rightarrow \\ (\mathsf{Phy}(\sigma) \ast \boxed{a} \vdash \mathsf{wp}_{\mathcal{E}}\, e\, \{x.\, \varphi(x)\}) \Rightarrow \\ \sigma; [e] \to^{*} \sigma'; T' \Rightarrow \\ \forall e' \in T'.\, \text{expr2val}(e') \neq \bot \lor \text{red}(e', \sigma') \end{array}$$

Notice that this is stronger than saying that the thread pool can reduce; we actually assert that *every* non-finished thread can take a step.

6 Model and semantics

The semantics closely follows the ideas laid out in [2].

6.1 Generic model of base logic

The base logic including equality, later, always, and a notion of ownership is defined on $\mathit{UPred}(M)$ for any CMRA M.

Interpretation of base assertions $\boxed{[\![\Gamma \vdash t : \mathsf{Prop}]\!] : [\![\Gamma]\!] \xrightarrow{\mathrm{ne}} \mathit{UPred}(M)}$

Remember that $\mathit{UPred}(M)$ is isomorphic to $M \xrightarrow{\mathrm{mon}} \mathit{SProp}$. We are thus going to define the assertions as mapping CMRA elements to sets of step-indices.

We introduce an additional logical connective $\mathsf{Own}(a)$, which will later be used to encode all of $\boxed{P}^{\iota}$, $[a]$ and $\mathsf{Phy}(\sigma)$.

$$
\begin{aligned}
[\![\Gamma \vdash t =_\tau u : \mathsf{Prop}]\!]_\gamma &\triangleq \lambda\_.\ \left\{ n \;\middle|\; [\![\Gamma \vdash t : \tau]\!]_\gamma \stackrel{n}{=} [\![\Gamma \vdash u : \tau]\!]_\gamma \right\} \\
[\![\Gamma \vdash \mathsf{False} : \mathsf{Prop}]\!]_\gamma &\triangleq \lambda\_.\ \emptyset \\
[\![\Gamma \vdash \mathsf{True} : \mathsf{Prop}]\!]_\gamma &\triangleq \lambda\_.\ \mathbb{N} \\
[\![\Gamma \vdash P \land Q : \mathsf{Prop}]\!]_\gamma &\triangleq \lambda a.\ [\![\Gamma \vdash P : \mathsf{Prop}]\!]_\gamma(a) \cap [\![\Gamma \vdash Q : \mathsf{Prop}]\!]_\gamma(a) \\
[\![\Gamma \vdash P \lor Q : \mathsf{Prop}]\!]_\gamma &\triangleq \lambda a.\ [\![\Gamma \vdash P : \mathsf{Prop}]\!]_\gamma(a) \cup [\![\Gamma \vdash Q : \mathsf{Prop}]\!]_\gamma(a) \\
[\![\Gamma \vdash P \Rightarrow Q : \mathsf{Prop}]\!]_\gamma &\triangleq \lambda a.\ \left\{ n \;\middle|\; \begin{aligned} &\forall m, b.\ m \leq n \land a \preccurlyeq b \land b \in \mathcal{V}_m \Rightarrow \\ &\qquad m \in [\![\Gamma \vdash P : \mathsf{Prop}]\!]_\gamma(b) \Rightarrow \\ &\qquad m \in [\![\Gamma \vdash Q : \mathsf{Prop}]\!]_\gamma(b) \end{aligned} \right\} \\
[\![\Gamma \vdash \forall x : \tau.\ P : \mathsf{Prop}]\!]_\gamma &\triangleq \lambda a.\ \left\{ n \;\middle|\; \forall v \in [\![\tau]\!].\ n \in [\![\Gamma, x : \tau \vdash P : \mathsf{Prop}]\!]_{\gamma[x \mapsto v]}(a) \right\} \\
[\![\Gamma \vdash \exists x : \tau.\ P : \mathsf{Prop}]\!]_\gamma &\triangleq \lambda a.\ \left\{ n \;\middle|\; \exists v \in [\![\tau]\!].\ n \in [\![\Gamma, x : \tau \vdash P : \mathsf{Prop}]\!]_{\gamma[x \mapsto v]}(a) \right\}
\end{aligned}
$$

$$
\begin{aligned}
[\![\Gamma \vdash \Box P : \mathsf{Prop}]\!]_\gamma &\triangleq \lambda a.\ [\![\Gamma \vdash P : \mathsf{Prop}]\!]_\gamma(|a|) \\
[\![\Gamma \vdash \triangleright P : \mathsf{Prop}]\!]_\gamma &\triangleq \lambda a.\ \{ n \mid n = 0 \lor n - 1 \in [\![\Gamma \vdash P : \mathsf{Prop}]\!]_\gamma(a) \} \\
[\![\Gamma \vdash P * Q : \mathsf{Prop}]\!]_\gamma &\triangleq \lambda a.\ \left\{ n \;\middle|\; \begin{aligned} &\exists b_1, b_2.\ a \stackrel{n}{=} b_1 \cdot b_2 \land \\ &\qquad n \in [\![\Gamma \vdash P : \mathsf{Prop}]\!]_\gamma(b_1) \land n \in [\![\Gamma \vdash Q : \mathsf{Prop}]\!]_\gamma(b_2) \end{aligned} \right\} \\
[\![\Gamma \vdash P \mathrel{-\!\!*} Q : \mathsf{Prop}]\!]_\gamma &\triangleq \lambda a.\ \left\{ n \;\middle|\; \begin{aligned} &\forall m, b.\ m \leq n \land a \cdot b \in \mathcal{V}_m \Rightarrow \\ &\qquad m \in [\![\Gamma \vdash P : \mathsf{Prop}]\!]_\gamma(b) \Rightarrow \\ &\qquad m \in [\![\Gamma \vdash Q : \mathsf{Prop}]\!]_\gamma(a \cdot b) \end{aligned} \right\} \\
[\![\Gamma \vdash \mathsf{Own}(a) : \mathsf{Prop}]\!]_\gamma &\triangleq \lambda b.\ \left\{ n \;\middle|\; [\![\Gamma \vdash a : \mathsf{M}]\!] \stackrel{n}{\preccurlyeq} b \right\} \\
[\![\Gamma \vdash \mathcal{V}(a) : \mathsf{Prop}]\!]_\gamma &\triangleq \lambda\_.\ \{ n \mid [\![\Gamma \vdash a : \tau]\!] \in \mathcal{V}_n \}
\end{aligned}
$$

For every definition, we have to show all the side-conditions: The maps have to be non-expansive and monotone.

6.2 Iris model

Semantic domain of assertions. The first complicated task in building a model of full Iris is defining the semantic model of Prop. We start by defining the functor that assembles the CMRAs we need to the global resource CMRA:

$$\mathit{ResF}(T^{\mathrm{op}}, T) \triangleq \left\{ w : \mathbb{N} \xrightharpoonup{\mathrm{fin}} \mathrm{A{\small G}}(\blacktriangleright T), \pi : \mathrm{E{\small X}}(\mathit{State})^?, g : \Sigma(T^{\mathrm{op}}, T) \right\}$$

Above, $M^?$ is the monoid obtained by adding a unit to M. (It's not a coincidence that we used the same notation for the range of the core; it's the same type either way: $M + 1$.) Remember that Σ is the user-chosen bifunctor from $\mathcal{COFE}$ to $\mathcal{CMRA}$ (see §5). $\mathit{ResF}(T^{\mathrm{op}}, T)$ is a CMRA by lifting the individual CMRAs pointwise. Furthermore, since Σ is locally contractive, so is ResF.

Now we can write down the recursive domain equation:

$$\mathit{iPreProp} \cong \mathit{UPred}(\mathit{ResF}(\mathit{iPreProp}, \mathit{iPreProp}))$$

$\mathit{iPreProp}$ is a COFE defined as the fixed-point of a locally contractive bifunctor. This fixed-point exists and is unique by America and Rutten's theorem [1, 3]. We do not need to consider how the object is constructed. We only need the isomorphism, given by

$$\begin{aligned} \mathit{Res} &\triangleq \mathit{ResF}(\mathit{iPreProp}, \mathit{iPreProp}) \\ \mathit{iProp} &\triangleq \mathit{UPred}(\mathit{Res}) \\ \xi &: \mathit{iProp} \xrightarrow{\mathrm{ne}} \mathit{iPreProp} \\ \xi^{-1} &: \mathit{iPreProp} \xrightarrow{\mathrm{ne}} \mathit{iProp} \end{aligned}$$

We then pick iProp as the interpretation of Prop:

$$[\![\mathsf{Prop}]\!] \triangleq \mathit{iProp}$$

Interpretation of assertions. iProp is a UPred, and hence the definitions from §6.1 apply. We only have to define the interpretation of the missing connectives, the most interesting bits being primitive view shifts and weakest preconditions.

World satisfaction $\boxed{- \models_{-} - : \Delta \mathit{State} \times \Delta\wp(\mathbb{N}) \times \mathit{Res} \xrightarrow{\mathrm{ne}} \mathit{SProp}}$

$$\begin{aligned} \mathit{pre\text{-}wsat}(n, \mathcal{E}, \sigma, R, r) \triangleq{}& r \in \mathcal{V}_{n+1} \land r.\pi = \mathsf{ex}(\sigma) \land \mathrm{dom}(R) \subseteq \mathcal{E} \cap \mathrm{dom}(r.w) \land \\ & \forall \iota \in \mathcal{E}, P \in \mathit{iProp}.\ (r.w)(\iota) \stackrel{n+1}{=} \mathsf{ag}(\mathsf{next}(\xi(P))) \Rightarrow n \in P(R(\iota)) \\ \sigma \models_{\mathcal{E}} r \triangleq{}& \{0\} \cup \left\{ n+1 \;\middle|\; \exists R : \mathbb{N} \xrightharpoonup{\mathrm{fin}} \mathit{Res}.\ \mathit{pre\text{-}wsat}(n, \mathcal{E}, \sigma, R, r \cdot \prod_{\iota} R(\iota)) \right\} \end{aligned}$$

Primitive view-shift $\boxed{\mathit{pvs}_{-}^{-}(-) : \Delta(\wp(\mathbb{N})) \times \Delta(\wp(\mathbb{N})) \times \mathit{iProp} \xrightarrow{\mathrm{ne}} \mathit{iProp}}$

$$\mathit{pvs}_{\mathcal{E}_1}^{\mathcal{E}_2}(P) = \lambda r. \left\{ n \;\middle|\; \begin{aligned} & \forall r_{\mathrm{f}}, k, \mathcal{E}_{\mathrm{f}}, \sigma.\ 0 < k \leq n \land (\mathcal{E}_1 \cup \mathcal{E}_2) \mathrel{\#} \mathcal{E}_{\mathrm{f}} \land k \in \sigma \models_{\mathcal{E}_1 \cup \mathcal{E}_{\mathrm{f}}} r \cdot r_{\mathrm{f}} \Rightarrow \\ & \qquad \exists s.\ k \in P(s) \land k \in \sigma \models_{\mathcal{E}_2 \cup \mathcal{E}_{\mathrm{f}}} s \cdot r_{\mathrm{f}} \end{aligned} \right\}$$

Weakest precondition $\boxed{\mathit{wp}_{-}(-, -) : \Delta(\wp(\mathbb{N})) \times \Delta(\mathit{Exp}) \times (\Delta(\mathit{Val}) \xrightarrow{\mathrm{ne}} \mathit{iProp}) \xrightarrow{\mathrm{ne}} \mathit{iProp}}$

wp is defined as the fixed-point of a contractive function.

$$\mathit{pre\text{-}wp}(\mathit{wp})(\mathcal{E}, e, \varphi) \triangleq \lambda r. \left\{ n \;\middle|\; \begin{aligned} & \forall r_{\mathrm{f}}, m, \mathcal{E}_{\mathrm{f}}, \sigma.\ 0 \leq m < n \land \mathcal{E} \mathrel{\#} \mathcal{E}_{\mathrm{f}} \land m + 1 \in \sigma \models_{\mathcal{E} \cup \mathcal{E}_{\mathrm{f}}} r \cdot r_{\mathrm{f}} \Rightarrow \\ & \quad (\forall v.\ \mathrm{expr2val}(e) = v \Rightarrow \exists s.\ m + 1 \in \varphi(v)(s) \land m + 1 \in \sigma \models_{\mathcal{E} \cup \mathcal{E}_{\mathrm{f}}} s \cdot r_{\mathrm{f}}) \land \\ & \quad (\mathrm{expr2val}(e) = \bot \land 0 < m \Rightarrow \mathrm{red}(e, \sigma) \land \forall e_2, \sigma_2, e_{\mathrm{f}}.\ e, \sigma \to e_2, \sigma_2, e_{\mathrm{f}} \Rightarrow \\ & \qquad\quad \exists s_1, s_2.\ m \in \sigma \models_{\mathcal{E} \cup \mathcal{E}_{\mathrm{f}}} s_1 \cdot s_2 \cdot r_{\mathrm{f}} \land m \in \mathit{wp}(\mathcal{E}, e_2, \varphi)(s_1) \land \\ & \qquad\qquad\quad (e_{\mathrm{f}} = \bot \lor m \in \mathit{wp}(\top, e_{\mathrm{f}}, \lambda\_.\ \lambda\_.\ \mathbb{N})(s_2)) \end{aligned} \right\}$$

$$\mathit{wp}_{\mathcal{E}}(e, \varphi) \triangleq \mathit{fix}(\mathit{pre\text{-}wp})(\mathcal{E}, e, \varphi)$$

Interpretation of program logic assertions $\boxed{[\![\Gamma \vdash t : \mathsf{Prop}]\!] : [\![\Gamma]\!] \xrightarrow{\text{ne}} \mathit{iProp}}$

$\boxed{P}^{\iota}$, $\boxed{a}$ and $\mathsf{Phy}(\sigma)$ are just syntactic sugar for forms of $\mathsf{Own}(-)$.

$$\begin{aligned}\boxed{P}^{\iota} &\triangleq \mathsf{Own}([\iota \mapsto \mathsf{ag}(\mathsf{next}(\xi(P)))], \varepsilon, \varepsilon)\\ \boxed{a} &\triangleq \mathsf{Own}(\varepsilon, \varepsilon, a)\\ \mathsf{Phy}(\sigma) &\triangleq \mathsf{Own}(\varepsilon, \mathsf{ex}(\sigma), \varepsilon)\end{aligned}$$

$$\begin{aligned}[\![\Gamma \vdash {}^{\mathcal{E}_1}\!\Rrightarrow^{\mathcal{E}_2} P : \mathsf{Prop}]\!]_\gamma &\triangleq \mathit{pvs}^{[\![\Gamma \vdash \mathcal{E}_2 : \mathsf{InvMask}]\!]_\gamma}_{[\![\Gamma \vdash \mathcal{E}_1 : \mathsf{InvMask}]\!]_\gamma}([\![\Gamma \vdash P : \mathsf{Prop}]\!]_\gamma)\\ [\![\Gamma \vdash \mathsf{wp}_{\mathcal{E}}\; e\; \{x.\, P\} : \mathsf{Prop}]\!]_\gamma &\triangleq \mathit{wp}_{[\![\Gamma \vdash \mathcal{E} : \mathsf{InvMask}]\!]_\gamma}([\![\Gamma \vdash e : \mathsf{Expr}]\!]_\gamma, \lambda v.\, [\![\Gamma \vdash P : \mathsf{Prop}]\!]_{\gamma[x \mapsto v]})\end{aligned}$$

Remaining semantic domains, and interpretation of non-assertion terms. The remaining domains are interpreted as follows:

$$\begin{array}{lll} [\![\mathsf{InvName}]\!] \triangleq \Delta\mathbb{N} & [\![\mathsf{Val}]\!] \triangleq \Delta\, \mathit{Val} & [\![1]\!] \triangleq \Delta\{()\}\\ [\![\mathsf{InvMask}]\!] \triangleq \Delta\wp(\mathbb{N}) & [\![\mathsf{Expr}]\!] \triangleq \Delta \mathit{Expr} & [\![\tau \times \tau']\!] \triangleq [\![\tau]\!] \times [\![\tau]\!]\\ [\![\mathsf{M}]\!] \triangleq F(\mathit{iProp}) & [\![\mathsf{State}]\!] \triangleq \Delta \mathit{State} & [\![\tau \to \tau']\!] \triangleq [\![\tau]\!] \xrightarrow{\text{ne}} [\![\tau]\!]\end{array}$$

For the remaining base types τ defined by the signature $\mathcal{S}$, we pick an object X_τ in $\mathcal{COFE}$ and define

$$[\![\tau]\!] \triangleq X_\tau$$

For each function symbol $F : \tau_1, \ldots, \tau_n \to \tau_{n+1} \in \mathcal{F}$, we pick a function $[\![F]\!] : [\![\tau_1]\!] \times \cdots \times [\![\tau_n]\!] \xrightarrow{\text{ne}} [\![\tau_{n+1}]\!]$.

Interpretation of non-propositional terms $\boxed{[\![\Gamma \vdash t : \tau]\!] : [\![\Gamma]\!] \xrightarrow{\text{ne}} [\![\tau]\!]}$

$$\begin{aligned}[\![\Gamma \vdash x : \tau]\!]_\gamma &\triangleq \gamma(x)\\ [\![\Gamma \vdash F(t_1, \ldots, t_n) : \tau_{n+1}]\!]_\gamma &\triangleq [\![F]\!]([\![\Gamma \vdash t_1 : \tau_1]\!]_\gamma, \ldots, [\![\Gamma \vdash t_n : \tau_n]\!]_\gamma)\\ [\![\Gamma \vdash \lambda x : \tau.\, t : \tau \to \tau']\!]_\gamma &\triangleq \lambda u : [\![\tau]\!].\, [\![\Gamma, x : \tau \vdash t : \tau]\!]_{\gamma[x \mapsto u]}\\ [\![\Gamma \vdash t(u) : \tau']\!]_\gamma &\triangleq [\![\Gamma \vdash t : \tau \to \tau']\!]_\gamma([\![\Gamma \vdash u : \tau]\!]_\gamma)\\ [\![\Gamma \vdash \mu x : \tau.\, t : \tau]\!]_\gamma &\triangleq \mathit{fix}(\lambda u : [\![\tau]\!].\, [\![\Gamma, x : \tau \vdash t : \tau]\!]_{\gamma[x \mapsto u]})\end{aligned}$$

$$\begin{aligned}[\![\Gamma \vdash () : 1]\!]_\gamma &\triangleq ()\\ [\![\Gamma \vdash (t_1, t_2) : \tau_1 \times \tau_2]\!]_\gamma &\triangleq ([\![\Gamma \vdash t_1 : \tau_1]\!]_\gamma, [\![\Gamma \vdash t_2 : \tau_2]\!]_\gamma)\\ [\![\Gamma \vdash \pi_i(t) : \tau_i]\!]_\gamma &\triangleq \pi_i([\![\Gamma \vdash t : \tau_1 \times \tau_2]\!]_\gamma)\end{aligned}$$

$$\begin{aligned}[\![\Gamma \vdash \varepsilon : \mathsf{M}]\!]_\gamma &\triangleq \varepsilon\\ [\![\Gamma \vdash |a| : \mathsf{M}]\!]_\gamma &\triangleq |[\![\Gamma \vdash a : \mathsf{M}]\!]_\gamma|\\ [\![\Gamma \vdash a \cdot b : \mathsf{M}]\!]_\gamma &\triangleq [\![\Gamma \vdash a : \mathsf{M}]\!]_\gamma \cdot [\![\Gamma \vdash b : \mathsf{M}]\!]_\gamma\end{aligned}$$

An environment Γ is interpreted as the set of finite partial functions ρ, with $\mathrm{dom}(\rho) = \mathrm{dom}(\Gamma)$ and $\rho(x) \in [\![\Gamma(x)]\!]$.

Logical entailment. We can now define *semantic* logical entailment.

Interpretation of entailment $\boxed{[\![\Gamma \mid \Theta \vdash P]\!] : \mathit{Prop}}$

$$[\![\Gamma \mid \Theta \vdash P]\!] \triangleq \forall n \in \mathbb{N}.\, \forall r \in \mathit{Res}.\, \forall \gamma \in [\![\Gamma]\!], \\ \big(\forall Q \in \Theta.\, n \in [\![\Gamma \vdash Q : \mathsf{Prop}]\!]_\gamma(r)\big) \Rightarrow n \in [\![\Gamma \vdash P : \mathsf{Prop}]\!]_\gamma(r)$$

The soundness statement of the logic reads

$$\Gamma \mid \Theta \vdash P \Rightarrow [\![\Gamma \mid \Theta \vdash P]\!]$$

7 Derived proof rules and other constructions

We will below abuse notation, using the *term* meta-variables like v to range over (bound) *variables* of the corresponding type. We omit type annotations in binders and equality, when the type is clear from context. We assume that the signature $\mathcal{S}$ embeds all the meta-level concepts we use, and their properties, into the logic. (The Coq formalization is a *shallow embedding* of the logic, so we have direct access to all meta-level notions within the logic anyways.)

7.1 Base logic

We collect here some important and frequently used derived proof rules.

$$P \Rightarrow Q \vdash P \mathrel{-\!\!*} Q \qquad P * \exists x.\, Q \dashv\vdash \exists x.\, P * Q \qquad P * \forall x.\, Q \vdash \forall x.\, P * Q \qquad \Box(P * Q) \dashv\vdash \Box P * \Box Q$$

$$\Box(P \Rightarrow Q) \vdash \Box P \Rightarrow \Box Q \qquad \Box(P \mathrel{-\!\!*} Q) \vdash \Box P \mathrel{-\!\!*} \Box Q \qquad \Box(P \mathrel{-\!\!*} Q) \dashv\vdash \Box(P \Rightarrow Q)$$

$$\triangleright(P \Rightarrow Q) \vdash \triangleright P \Rightarrow \triangleright Q \qquad \triangleright(P \mathrel{-\!\!*} Q) \vdash \triangleright P \mathrel{-\!\!*} \triangleright Q \qquad \frac{\Theta, \triangleright P \vdash P}{\Theta \vdash P}$$

Persistent assertions.

Definition 19. *An assertion P is* persistent *if $P \vdash \Box P$.*

Of course, $\Box P$ is persistent for any P. Furthermore, by the proof rules given in §5.3, $t = t'$ as well as $\boxed{|a|}$, $\mathcal{V}(a)$ and $\boxed{P}^{\iota}$ are persistent. Persistence is preserved by conjunction, disjunction, separating conjunction as well as universal and existential quantification.

In our proofs, we will implicitly add and remove $\Box$ from persistent assertions as necessary, and generally treat them like normal, non-linear assumptions.

Timeless assertions. We can show that the following additional closure properties hold for timeless assertions:

$$\frac{\Gamma \vdash \mathsf{timeless}(P) \qquad \Gamma \vdash \mathsf{timeless}(Q)}{\Gamma \vdash \mathsf{timeless}(P \land Q)} \qquad \frac{\Gamma \vdash \mathsf{timeless}(P) \qquad \Gamma \vdash \mathsf{timeless}(Q)}{\Gamma \vdash \mathsf{timeless}(P \lor Q)}$$

$$\frac{\Gamma \vdash \mathsf{timeless}(P) \qquad \Gamma \vdash \mathsf{timeless}(Q)}{\Gamma \vdash \mathsf{timeless}(P * Q)} \qquad \frac{\Gamma \vdash \mathsf{timeless}(P)}{\Gamma \vdash \mathsf{timeless}(\Box P)}$$

7.2 Program logic

Hoare triples and view shifts are syntactic sugar for weakest (liberal) preconditions and primitive view shifts, respectively:

$$\{P\}\, e\, \{v.\, Q\}_{\mathcal{E}} \triangleq \Box(P \Rightarrow \mathsf{wp}_{\mathcal{E}}\, e\, \{\lambda v.\, Q\}) \qquad \begin{aligned} P \;{}^{\mathcal{E}_1}\!\!\Rrightarrow^{\mathcal{E}_2} Q &\triangleq \Box(P \Rightarrow {}^{\mathcal{E}_1}\!\!\models\!\!\Rrightarrow^{\mathcal{E}_2} Q) \\ P \;{}^{\mathcal{E}_1}\!\!\Lleftarrow\!\!\Rrightarrow^{\mathcal{E}_2} Q &\triangleq P \;{}^{\mathcal{E}_1}\!\!\Rrightarrow^{\mathcal{E}_2} Q \land Q \;{}^{\mathcal{E}2}\!\!\Rrightarrow^{\mathcal{E}_1} P \end{aligned}$$

We write just one mask for a view shift when $\mathcal{E}_1 = \mathcal{E}_2$. Clearly, all of these assertions are persistent. The convention for omitted masks is similar to the base logic: An omitted $\mathcal{E}$ is $\top$ for Hoare triples and $\emptyset$ for view shifts.

View shifts. The following rules can be derived for view shifts.

$$\textsc{vs-update}\quad \frac{a \rightsquigarrow B}{\boxed{a} \Rrightarrow \exists b \in B.\, \boxed{b}}
\qquad
\textsc{vs-trans}\quad \frac{P \;{}^{\mathcal{E}_1}{\Rrightarrow}^{\mathcal{E}_2}\; Q \qquad Q \;{}^{\mathcal{E}_2}{\Rrightarrow}^{\mathcal{E}_3}\; R \qquad \mathcal{E}_2 \subseteq \mathcal{E}_1 \cup \mathcal{E}_3}{P \;{}^{\mathcal{E}_1}{\Rrightarrow}^{\mathcal{E}_3}\; R}
\qquad
\textsc{vs-imp}\quad \frac{\Box(P \Rightarrow Q)}{P \Rrightarrow_{\emptyset} Q}$$

$$\textsc{vs-mask-frame}\quad \frac{P \;{}^{\mathcal{E}_1}{\Rrightarrow}^{\mathcal{E}_2}\; Q}{P \;{}^{\mathcal{E}_1 \uplus \mathcal{E}'}{\Rrightarrow}^{\mathcal{E}_2 \uplus \mathcal{E}'}\; Q}
\qquad
\textsc{vs-frame}\quad \frac{P \;{}^{\mathcal{E}_1}{\Rrightarrow}^{\mathcal{E}_2}\; Q}{P * R \;{}^{\mathcal{E}_1}{\Rrightarrow}^{\mathcal{E}_2}\; Q * R}
\qquad
\textsc{vs-timeless}\quad \frac{\mathsf{timeless}(P)}{\triangleright P \Rrightarrow P}
\qquad
\textsc{vs-allocI}\quad \frac{\mathsf{infinite}(\mathcal{E})}{\triangleright P \Rrightarrow_{\mathcal{E}} \exists \iota \in \mathcal{E}.\, \boxed{P}^{\iota}}$$

$$\textsc{vs-openI}\quad \boxed{P}^{\iota} \vdash \mathsf{True} \;{}^{\{\iota\}}{\Rrightarrow}^{\emptyset}\; \triangleright P
\qquad
\textsc{vs-closeI}\quad \boxed{P}^{\iota} \vdash \triangleright P \;{}^{\emptyset}{\Rrightarrow}^{\{\iota\}}\; \mathsf{True}
\qquad
\textsc{vs-disj}\quad \frac{P \;{}^{\mathcal{E}_1}{\Rrightarrow}^{\mathcal{E}_2}\; R \qquad Q \;{}^{\mathcal{E}_1}{\Rrightarrow}^{\mathcal{E}_2}\; R}{P \vee Q \;{}^{\mathcal{E}_1}{\Rrightarrow}^{\mathcal{E}_2}\; R}$$

$$\textsc{vs-exist}\quad \frac{\forall x.\, (P \;{}^{\mathcal{E}_1}{\Rrightarrow}^{\mathcal{E}_2}\; Q)}{(\exists x.\, P) \;{}^{\mathcal{E}_1}{\Rrightarrow}^{\mathcal{E}_2}\; Q}
\qquad
\textsc{vs-box}\quad \frac{\Box Q \vdash P \;{}^{\mathcal{E}_1}{\Rrightarrow}^{\mathcal{E}_2}\; R}{P \wedge \Box Q \;{}^{\mathcal{E}_1}{\Rrightarrow}^{\mathcal{E}_2}\; R}
\qquad
\textsc{vs-false}\quad \mathsf{False} \;{}^{\mathcal{E}_1}{\Rrightarrow}^{\mathcal{E}_2}\; P$$

Hoare triples. The following rules can be derived for Hoare triples.

$$\textsc{Ht-ret}\quad \{\mathsf{True}\}\; w\; \{v.\, v = w\}_{\mathcal{E}}
\qquad
\textsc{Ht-bind}\quad \frac{K \text{ is a context} \qquad \{P\}\; e\; \{v.\, Q\}_{\mathcal{E}} \qquad \forall v.\, \{Q\}\; K(v)\; \{w.\, R\}_{\mathcal{E}}}{\{P\}\; K(e)\; \{w.\, R\}_{\mathcal{E}}}$$

$$\textsc{Ht-csq}\quad \frac{P \Rrightarrow P' \qquad \{P'\}\; e\; \{v.\, Q'\}_{\mathcal{E}} \qquad \forall v.\, Q' \Rrightarrow Q}{\{P\}\; e\; \{v.\, Q\}_{\mathcal{E}}}
\qquad
\textsc{Ht-mask-weaken}\quad \frac{\{P\}\; e\; \{v.\, Q\}_{\mathcal{E}}}{\{P\}\; e\; \{v.\, Q\}_{\mathcal{E} \uplus \mathcal{E}'}}$$

$$\textsc{Ht-frame}\quad \frac{\{P\}\; e\; \{v.\, Q\}_{\mathcal{E}}}{\{P * R\}\; e\; \{v.\, Q * R\}_{\mathcal{E}}}
\qquad
\textsc{Ht-frame-step}\quad \frac{\begin{array}{c}\{P\}\; e\; \{v.\, Q\}_{\mathcal{E}} \qquad \mathrm{expr2val}(e) = \bot \qquad \mathcal{E}_2 \subseteq \mathcal{E}_2 \\ R_1 \;{}^{\mathcal{E}_1}{\Rrightarrow}^{\mathcal{E}_2}\; \triangleright R_2 \qquad R_2 \;{}^{\mathcal{E}_2}{\Rrightarrow}^{\mathcal{E}_1}\; R_3\end{array}}{\{P * R_1\}\; e\; \{v.\, Q * R_3\}_{\mathcal{E} \uplus \mathcal{E}_1}}$$

$$\textsc{Ht-atomic}\quad \frac{P \;{}^{\mathcal{E} \uplus \mathcal{E}'}{\Rrightarrow}^{\mathcal{E}}\; P' \qquad \{P'\}\; e\; \{v.\, Q'\}_{\mathcal{E}} \qquad \forall v.\, Q' \;{}^{\mathcal{E}}{\Rrightarrow}^{\mathcal{E} \uplus \mathcal{E}'}\; Q \qquad \mathsf{atomic}(\mathrm{e})}{\{P\}\; e\; \{v.\, Q\}_{\mathcal{E} \uplus \mathcal{E}'}}$$

$$\textsc{Ht-disj}\quad \frac{\{P\}\; e\; \{v.\, R\}_{\mathcal{E}} \qquad \{Q\}\; e\; \{v.\, R\}_{\mathcal{E}}}{\{P \vee Q\}\; e\; \{v.\, R\}_{\mathcal{E}}}
\qquad
\textsc{Ht-exist}\quad \frac{\forall x.\, \{P\}\; e\; \{v.\, Q\}_{\mathcal{E}}}{\{\exists x.\, P\}\; e\; \{v.\, Q\}_{\mathcal{E}}}
\qquad
\textsc{Ht-box}\quad \frac{\Box Q \vdash \{P\}\; e\; \{v.\, R\}_{\mathcal{E}}}{\{P \wedge \Box Q\}\; e\; \{v.\, R\}_{\mathcal{E}}}$$

$$\textsc{Ht-false}\quad \{\mathsf{False}\}\; e\; \{v.\, P\}_{\mathcal{E}}
\qquad
\textsc{Ht-inv}\quad \frac{\{\triangleright R * P\}\; e\; \{v.\, \triangleright R * Q\}_{\mathcal{E}} \qquad \mathsf{atomic}(\mathrm{e})}{\boxed{R}^{\iota} \vdash \{P\}\; e\; \{v.\, Q\}_{\mathcal{E} \uplus \{\iota\}}}$$

$$\textsc{Ht-inv-timeless}\quad \frac{\{R * P\}\; e\; \{v.\, R * Q\}_{\mathcal{E}} \qquad \mathsf{atomic}(\mathrm{e}) \qquad \mathsf{timeless}(R)}{\boxed{R}^{\iota} \vdash \{P\}\; e\; \{v.\, Q\}_{\mathcal{E} \uplus \{\iota\}}}$$

Lifting of operational semantics. We can derive some specialized forms of the lifting axioms for the operational semantics.

WP-LIFT-ATOMIC-STEP

$$\frac{\text{atomic}(e_1) \qquad \text{red}(e_1, \sigma_1)}{\begin{array}{l} \triangleright\mathsf{Phy}(\sigma_1) * \triangleright\forall v_2, \sigma_2, e_\text{f}.\, (e_1, \sigma_1 \to \text{val2expr}(v), \sigma_2, e_\text{f}) \land \mathsf{Phy}(\sigma_2) \mathrel{-\!\!*} P[v_2/x] * \mathsf{wp}_\top\, e_\text{f}\, \{\_.\, \mathsf{True}\} \\ \vdash \mathsf{wp}_{\mathcal{E}_1}\, e_1\, \{x.\, P\} \end{array}}$$

WP-LIFT-ATOMIC-DET-STEP

$$\frac{\text{atomic}(e_1) \qquad \text{red}(e_1, \sigma_1) \qquad \forall e_2', \sigma_2', e_\text{f}'.\, e_1, \sigma_1 \to e_2, \sigma_2, e_\text{f} \Rightarrow \sigma_2 = \sigma_2' \land \text{expr2val}(e_2') = v_2 \land e_\text{f} = e_\text{f}'}{\triangleright\mathsf{Phy}(\sigma_1) * \triangleright(\mathsf{Phy}(\sigma_2) \mathrel{-\!\!*} P[v_2/x] * \mathsf{wp}_\top\, e_\text{f}\, \{\_.\, \mathsf{True}\}) \vdash \mathsf{wp}_{\mathcal{E}_1}\, e_1\, \{x.\, P\}}$$

WP-LIFT-PURE-DET-STEP

$$\frac{\text{expr2val}(e_1) = \bot \qquad \forall \sigma_1.\, \text{red}(e_1, \sigma_1) \qquad \forall \sigma_1, e_2', \sigma_2, e_\text{f}'.\, e_1, \sigma_1 \to e_2, \sigma_2, e_\text{f} \Rightarrow \sigma_1 = \sigma_2 \land e_2 = e_2' \land e_\text{f} = e_\text{f}'}{\triangleright(\mathsf{wp}_{\mathcal{E}_1}\, e_2\, \{x.\, P\} * \mathsf{wp}_\top\, e_\text{f}\, \{\_.\, \mathsf{True}\}) \vdash \mathsf{wp}_{\mathcal{E}_1}\, e_1\, \{x.\, P\}}$$

7.3 Global functor and ghost ownership

Hereinafter we assume the global CMRA functor (served up as a parameter to Iris) is obtained from a family of functors $(\Sigma_i)_{i \in I}$ for some finite I by picking

$$\Sigma(T) \triangleq \prod_{i \in I} \mathsf{GhName} \xrightarrow{\text{fin}} \Sigma_i(T)$$

We don't care so much about what concretely GhName is, as long as it is countable and infinite. With $M_i \triangleq \Sigma_i(\mathit{iProp})$, we write $\boxed{a : M_i}^{\gamma}$ (or just $\boxed{a}^{\gamma}$ if M_i is clear from the context) for $\boxed{[i \mapsto [\gamma \mapsto a]]}$. In other words, $\boxed{a : M_i}^{\gamma}$ asserts that in the current state of monoid M_i, the "ghost location" γ is allocated and we own piece a.

From PVS-UPDATE, VS-UPDATE and the frame-preserving updates in §3.1 and §3.3, we have the following derived rules.

GHOST-ALLOC-STRONG

$$\frac{G \text{ infinite}}{\mathsf{True} \Rrightarrow \exists \gamma \in G.\, \boxed{a : M_i}^{\gamma}}$$

GHOST-ALLOC

$$\mathsf{True} \Rrightarrow \exists \gamma.\, \boxed{a : M_i}^{\gamma}$$

GHOST-UPDATE

$$\frac{a \rightsquigarrow_{M_i} B}{\boxed{a : M_i}^{\gamma} \Rrightarrow \exists b \in B.\, \boxed{b : M_i}^{\gamma}}$$

GHOST-OP

$$\boxed{a : M_i}^{\gamma} * \boxed{b : M_i}^{\gamma} \Leftrightarrow \boxed{a \cdot b : M_i}^{\gamma}$$

GHOST-VALID

$$\boxed{a : M_i}^{\gamma} \Rightarrow \mathcal{V}_{M_i}(a)$$

GHOST-TIMELESS

$$\frac{a \text{ is a discrete COFE element}}{\mathsf{timeless}(\boxed{a : M_i}^{\gamma})}$$

7.4 Invariant identifier namespaces

Let $\mathcal{N} \in \mathsf{InvNamesp} \triangleq \mathsf{list}(\mathsf{InvName})$ be the type of *namespaces* for invariant names. Notice that there is an injection $\mathsf{namesp\_inj} : \mathsf{InvNamesp} \to \mathsf{InvName}$. Whenever needed (in particular, for masks at view shifts and Hoare triples), we coerce $\mathcal{N}$ to its suffix-closure:

$$\mathcal{N}^{\uparrow} \triangleq \{\iota \mid \exists \mathcal{N}'.\, \iota = \mathsf{namesp\_inj}(\mathcal{N}' +\!\!+ \mathcal{N})\}$$

We use the notation $\mathcal{N}.\iota$ for the namespace $[\iota] +\!\!+ \mathcal{N}$.

We define the inclusion relation on namespaces as $\mathcal{N}_1 \sqsubseteq \mathcal{N}_2 \Leftrightarrow \exists \mathcal{N}_3.\, \mathcal{N}_2 = \mathcal{N}_3 +\!\!+ \mathcal{N}_1$, *i.e.,* $\mathcal{N}_1$ is a suffix of $\mathcal{N}_2$. We have that $\mathcal{N}_1 \sqsubseteq \mathcal{N}_2 \Rightarrow \mathcal{N}_2^{\uparrow} \subseteq \mathcal{N}_1^{\uparrow}$.

Similarly, we define $\mathcal{N}_1 \mathrel{\#} \mathcal{N}_2 \triangleq \exists \mathcal{N}_1', \mathcal{N}_2'.\, \mathcal{N}_1' \sqsubseteq \mathcal{N}_1 \land \mathcal{N}_2' \sqsubseteq \mathcal{N}_2 \land |\mathcal{N}_1'| = |\mathcal{N}_2'| \land \mathcal{N}_1' \neq \mathcal{N}_2'$, *i.e.,* there exists a distinguishing suffix. We have that $\mathcal{N}_1 \mathrel{\#} \mathcal{N}_2 \Rightarrow \mathcal{N}_2^{\uparrow} \mathrel{\#} \mathcal{N}_1^{\uparrow}$, and furthermore $\iota_1 \neq \iota_2 \Rightarrow \mathcal{N}.\iota_1 \mathrel{\#} \mathcal{N}.\iota_2$.

We will overload the usual Iris notation for invariant assertions in the following:

$$\boxed{P}^{\mathcal{N}} \triangleq \exists \iota \in \mathcal{N}^{\uparrow}.\, \boxed{P}^{\iota}$$

We can now derive the following rules for this derived form of the invariant assertion:

$$\boxed{P}^{\mathcal{N}} \vdash \Box \boxed{P}^{\mathcal{N}} \qquad\qquad \triangleright P \vdash \mathrel{{\mid}\!{\Rrightarrow}}_{\mathcal{N}} \boxed{P}^{\mathcal{N}}$$

$$\frac{\mathsf{atomic}(\mathrm{e}) \qquad \mathcal{N} \subseteq \mathcal{E} \qquad \Theta \vdash \boxed{P}^{\mathcal{N}} \qquad \Theta \vdash \triangleright P \mathrel{-\!\!*} \mathsf{wp}_{\mathcal{E}\setminus\mathcal{N}}\, e\, \{v.\, \triangleright P * Q\}}{\Theta \vdash \mathsf{wp}_{\mathcal{E}}\, e\, \{v.\, Q\}}$$

$$\frac{\mathcal{N} \subseteq \mathcal{E} \qquad \Theta \vdash \boxed{P}^{\mathcal{N}} \qquad \Theta \vdash \triangleright P \mathrel{-\!\!*} \mathrel{{\mid}\!{\Rrightarrow}}_{\mathcal{E}\setminus\mathcal{N}} \triangleright P * Q}{\Theta \vdash \mathrel{{\mid}\!{\Rrightarrow}}_{\mathcal{E}} Q}$$

$$\frac{\mathsf{atomic}(\mathrm{e}) \qquad \mathcal{N} \subseteq \mathcal{E} \qquad \{\triangleright P * Q\}\, e\, \{v.\, \triangleright P * R\}_{\mathcal{E}\setminus\mathcal{N}}}{\boxed{P}^{\mathcal{N}} \vdash \{Q\}\, e\, \{v.\, R\}_{\mathcal{E}}} \qquad\qquad \frac{\mathcal{N} \subseteq \mathcal{E} \qquad \triangleright P * Q \Rrightarrow_{\mathcal{E}\setminus\mathcal{N}} \triangleright P * R}{\boxed{P}^{\mathcal{N}} \vdash Q \Rrightarrow_{\mathcal{E}} R}$$

References

[1] Pierre America and Jan Rutten. "Solving Reflexive Domain Equations in a Category of Complete Metric Spaces". In: *JCSS* 39.3 (1989), pp. 343–375.

[2] Lars Birkedal and Aleš Bizjak. *A Taste of Categorical Logic — Tutorial Notes*. Available at http://users-cs.au.dk/birke/modures/tutorial/categorical-logic-tutorial-notes.pdf. Oct. 2014.

[3] Lars Birkedal, Kristian Støvring, and Jacob Thamsborg. "The category-theoretic solution of recursive metric-space equations". In: *TCS* 411.47 (2010), pp. 4102–4122. DOI: 10.1016/j.tcs.2010.07.010. URL: http://dx.doi.org/10.1016/j.tcs.2010.07.010.

[4] Aaron Turon, Derek Dreyer, and Lars Birkedal. "Unifying refinement and Hoare-style reasoning in a logic for higher-order concurrency". In: *ICFP*. 2013, pp. 377–390.