CoLoSL: Concurrent Local Subjective Logic

Azalea Raad, Jules Villard, Philippa Gardner

January 3, 2015

Contents

1. CoLoSL Model and Assertions

We formally describe the underlying model of CoLoSL; we present the various ingredients necessary for defining the CoLoSL *worlds*: the building blocks of CoLoSL that track the resources held by each thread, the shared resources accessible to all threads, as well as the ways in which the shared resources may be manipulated by each thread.

We then proceed with the *assertions* of CoLoSL and provide their semantics by relating them to sets of worlds. Finally, we establish the validity of Copy, Forget and Merge principles introduced in the preceding chapters by establishing their truth for all possible worlds and interpretations.

1.1 Worlds

Overview A *world* is a triple $(l, g, \mathcal{I})$ that is *well-formed* where l and g are *logical states* and $\mathcal{I}$ is an *action model*; let us explain the role of each component informally. The *local logical state*, or simply local state, l represents the locally owned resources of a thread. The *shared logical state*, or shared state, g represents the *entire* (global) shared state, accessible to all threads, subject to interferences as described by the action models.

An action model is a partial function from *capabilities* to sets of *actions*. An action is a triple (p, q, c) of logical states where p and q are the *pre* and *post-states* of the action, respectively, and c is the action *condition*. That is, c acts as a mere catalyst for the action: it has to be present for the action to take effect, but is left unchanged by the action. The *action model* $\mathcal{I}$ corresponds directly to the (semantic interpretation of) an interference assertion I. Although worlds do not put further constraints on the relationship between $\mathcal{I}$ and g, they are linked more tightly in the semantics of assertions (§1.2).

Finally, the composition of two worlds will be defined whenever their local states are disjoint and they agree on all other two components, hence have identical knowledge of the shared state and possible interferences.

We proceed by defining logical states, which are CoLoSL's notion of *resource*, in the standard separation logic sense. Logical states have two components: one describes machine states (*e.g.* stacks and heaps); the other represents *capabilities*. The latter are inspired by the capabilities in deny-guarantee reasoning [7]: a thread in possession of a given capability is allowed to perform the associated actions (as prescribed by the *action model* components of each world, defined below), while any capability *not* owned by a thread means that the environment can perform the action.

CoLoSL is parametric in the choice of the separation algebra representing the machine states and capabilities. This allows for suitable *instantiation* of CoLoSL depending on the programs being verified. For instance, in the token ring example of §**??** the separation algebra of machine states is a standard variable stack; while capabilities are captured as a set of tokens. However, as we demonstrate in the examples of §**??**, our programs often call for a more complex model of machine states and capabilities. For instance, we may need our capabilities to be fractionally owned, where ownership of a *fraction* of a capability grants the right to perform the action to both the thread and the environment, while a fully-owned capability by the thread *denies* the right to the environment to perform the associated action.

In general, the separation algebra of machine states and capabilities can be instantiated with *any* separation algebra (*i.e.* a cancellative, partial commutative monoid [2]) that satisfies the *cross-split* property. This is formalised in the following parametrisations.

Parameter 1 (Machine states separation algebra)**.** Let $(\mathbb{M}, \bullet_{\mathbb{M}}, \mathbf{0}_{\mathbb{M}})$ be any separation algebra with the cross-split property, representing machine states where the elements of $\mathbb{M}$ are ranged over by $m, m_1, \cdots, m_n$.

Parameter 2 (Capability Separation Algebra)**.** Let $(\mathbb{K}, \bullet_{\mathbb{K}}, \mathbf{0}_{\mathbb{K}})$ be any separation algebra with the cross-split property, representing capability resources where the elements of $\mathbb{K}$ are ranged over by $\kappa, \kappa_1, \cdots \kappa_n$.

We can now formalise the notion of *logical states*. As discussed above, a logical state is a pair comprising a machine state and a capability resource.

Definition 1 (Logical states)**.** Given the separation algebra of machine states, $(\mathbb{M}, \bullet_{\mathbb{M}}, \mathbf{0}_{\mathbb{M}})$, and the separation algebra of capabilities, $(\mathbb{K}, \bullet_{\mathbb{K}}, \mathbf{0}_{\mathbb{K}})$, a *logical state* is a pair (m, κ), consisting of a machine state $m \in \mathbb{M}$ and a capability $\kappa \in \mathbb{K}$.

$$\mathsf{LState} \overset{\text{def}}{=} \mathbb{M} \times \mathbb{K}$$

We write $l, l_1, \cdots, l_n$ to range over either arbitrary logical states or those representing the local logical state. Similarly, we write $g, g_1, \cdots, g_n$ to range over logical states when representing the shared (global) state. We write $\mathbf{0}_{\mathsf{L}}$ for the logical state $(\mathbf{0}_{\mathbb{M}}, \mathbf{0}_{\mathbb{K}})$. Given a logical state l, we write l_{M} and l_{K} for the first and second projections, respectively. The *composition of logical states* $\circ : \mathsf{LState} \times \mathsf{LState} \rightharpoonup \mathsf{LState}$ is defined component-wise:

$$(m, \kappa) \circ (m', \kappa') \stackrel{\text{def}}{=} (m \bullet_{\mathbb{M}} m', \kappa \bullet_{\mathbb{K}} \kappa')$$

The *separation algebra of logical states* is given by $(\mathsf{LState}, \circ, \mathbf{0}_{\mathsf{L}})$.

Oftentimes, we need to compare two logical states $l_1 \leq l_2$ (or their constituents: $\kappa_1 \leq \kappa_2$, $m_1 \leq m_2$) defined when there exists l such that $l \circ l_1 = l_2$. This is captured in the following definition.

Definition 2 (Ordering)**.** Given any separation algebra $(\mathbb{B}, \bullet_{\mathbb{B}}, \mathbf{0}_{\mathbb{B}})$, the *ordering relation*, $\leq : \mathbb{B} \times \mathbb{B}$, is defined as:

$$\leq \stackrel{\text{def}}{=} \{(b_1, b_2) \mid \exists b.\, b_1 \bullet_{\mathbb{B}} b = b_2\}$$

We write $b_1 \leq b_2$ for $(b_1, b_2) \in \leq$. We also write $b_2 - b_1$ to denote the unique (by cancellativity of separation algebras) element in $\mathbb{B}$ such that $b_1 \bullet_{\mathbb{B}} (b_2 - b_1) = b_2$.

In our formalisms we occasionally need to quantify over *compatible* logical states (similarly, compatible machine states or compatible capabilities), *i.e.* those that can be composed together by $\circ$. Additionally, we often describe two logical states as *disjoint*. We formalise these notions below.

Definition 3 (Compatibility)**.** Given any separation algebra $(\mathbb{B}, \bullet_{\mathbb{B}}, \mathbf{0}_{\mathbb{B}})$, the *compatibility relation*, $\sharp : \mathbb{B} \times \mathbb{B}$, is defined as:

$$\sharp \stackrel{\text{def}}{=} \{(b_1, b_2) \mid \exists b.\, b_1 \bullet_{\mathbb{B}} b_2 = b\}$$

We write $b_1 \sharp b_2$ for $(b_1, b_2) \in \sharp$.

Definition 4 (Disjointness)**.** Given any separation algebra $(\mathbb{B}, \bullet_{\mathbb{B}}, \mathbf{0}_{\mathbb{B}})$, the *disjointness relation*, $\perp : \mathbb{B} \times \mathbb{B}$, is defined as:

$$\perp \stackrel{\text{def}}{=} \{(b_1, b_2) \mid b_1 \sharp b_2 \wedge \forall b \in \mathbb{B}.\, b \leq b_1 \wedge b \leq b_2 \implies b = \mathbf{0}_{\mathbb{B}}\}$$

We write $b_1 \perp b_2$ for $(b_1, b_2) \in \perp$.

Observe that for a separation algebra $(\mathbb{B}, \bullet_{\mathbb{B}}, \mathbf{0}_{\mathbb{B}})$ satisfying the disjointness property[1], the definitions of compatibility and disjointness relations coincide.

We now proceed with the next ingredients of a CoLoSL world, namely, action models. Recall from above that an action is simply a pair of logical states describing the pre- and post-states of the action, while an action model describes the set of actions associated with each capability.

Definition 5 (Actions, action models)**.** The set of *actions*, Action, ranged over by $a, a_1, \cdots, a_n$, is as defined below.

$$\mathsf{Action} \overset{\text{def}}{=} \mathsf{LState} \times \mathsf{LState} \times \mathsf{LState}$$

The set of *action models*, AMod, is defined as follows.

$$\mathsf{AMod} \overset{\text{def}}{=} \mathbb{K} \rightharpoonup \mathcal{P}(\mathsf{Action})$$

We write $\mathcal{I}, \mathcal{I}_1, \cdots, \mathcal{I}_n$ to range over action models; we write $\emptyset$ for an action model with empty domain.

The Effect of Actions Given a world $(l, g, \mathcal{I})$, since g represents the *entire* shared state, as part of the well-formedness condition of worlds we require that the actions in $\mathcal{I}$ are *confined* to g. Let us elaborate on the necessity of the confinement condition.

As threads may unilaterally decide to introduce part of their local states into the shared state at any point (by EXTEND), confinement ensures that existing actions cannot affect future extensions of the shared state. Similarly, we require that the new actions associated with newly shared state are confined to that extension in the same vein, hence extending the shared state cannot retroactively invalidate the views of other threads. However, as we will demonstrate, confinement does not prohibit *referring* to existing parts of the shared state in the new actions; rather, it only safeguards against *mutation* of the already shared resources through new actions.

Through confinement we ensure that the *effect* of actions in the local and global action models are confined to the shared state. In other words, given an action $a = (p, q, c)$ and a shared state g, whenever $p \circ c$ *agrees* with g then p must be contained in g. Agreement of $p \circ c$ and g merely means that they agree on the resources they have in common. In particular, g only needs to contain p (and not $p \circ c$) for a to take effect. This relaxation is due

[1]$\forall b, b' \in \mathbb{B}.\, b \bullet_{\mathbb{B}} b = b' \implies b = b' = \mathbf{0}_{\mathbb{B}}$

to the fact that other threads may extend the shared state; in particular, the extension may provide the missing resources for $p \circ c$ to be contained in the shared state, thus allowing the extending thread to perform action a. Crucially, however, the part of the shared state mutated by the action, namely p, must be contained in g so that extensions of the shared state need not worry about existing actions interfering with new resources that were never shared beforehand.

The agreement of two logical states (e.g. $p \circ c$ and g above) will be defined using the following notion of *intersection* of logical states. This will enable us to define our confinement condition.

Definition 6 (Intersection)**.** The *intersection* function over logical states, $\sqcap : (\mathsf{LState} \times \mathsf{LState}) \to \mathcal{P}(\mathsf{LState})$, is defined as follows.

$$l_1 \sqcap l_2 \overset{\text{def}}{=} \left\{ l \mid \exists l', l'_1, l'_2.\ l_1 = l \circ l'_1 \land l_2 = l \circ l'_2 \land l \circ l'_1 \circ l'_2 = l' \right\}$$

Observe that when the separation algebra of logical states satisfies the disjointness property, $l_1 \sqcap l_2$ yields at most one element for any l_1 and l_2.

Two logical states l_1 and l_2 then *agree* if their intersection is non-empty, *i.e.* $l_1 \sqcap l_2 \neq \emptyset$. We can now define action confinement.

Definition 7 (Action confinement)**.** An action $a = (p, q, c)$ is *confined* to a logical state g, written g © a, if for all r compatible with g ($g \sharp r$):

$$p \circ c \sqcap g \neq \emptyset \Rightarrow p \leq g \land p \perp r$$

As discussed, only the action pre-state p, i.e. the part actually mutated by the action has to be contained in g and must be disjoint from all potential extensions (r) of the logical state g. That is, future extensions of g need not account for existing actions interfering with new resources.

Given a shared state g and an action model $\mathcal{I}$, we require that all actions of $\mathcal{I}$ are confined in all possible *futures* of g, *i.e.* all shared states resulting from g after any number of applications of actions in $\mathcal{I}$. For that we define *action application* that describes the effect of an action on a logical state. Moreover, for some of the actions in $\mathcal{I}$, the pre-state may not affect g, *i.e.* its intersection with g may be the empty state $\mathbf{0}_{\mathsf{L}}$. In that case, we find that even though that action is potentially enabled, we do not need to account for it since it leaves g unchanged. We thus introduce the notion of *visible actions* to quantify over those actions that affect (mutate) g.

Definition 8 (Action application)**.** The *application* of an action $a = (p, q, c)$ on a logical state g, written $a[g]$, is defined provided that there exists l such that

$$p \circ c \sqcap g \neq \emptyset \wedge g = p \circ l \wedge q \sharp l$$

When that is the case, we write $a[g]$ for the (uniquely defined) logical state $q \circ l$. We write $\mathit{potential}(a, g)$ to denote that $a[g]$ is defined.

Definition 9 (Visible actions)**.** An action $a = (p, q, c)$ is called *visible in* g, written $\mathit{visible}(a, g)$ when

$$\exists l \in (p \sqcap g) \,.\, l \neq \mathbf{0}_{\mathsf{L}}$$

We are now ready to define our confinement condition on action models. Inspired by Local RG [8], we introduce the concept of locally fenced action models to capture all possible states reachable from the current state via some number of action applications. A set of states $\mathcal{F}$ *fences* an action model if it is invariant under interferences perpetrated by the corresponding actions. An action model is then confined to a logical state l if it can be fenced by a set of states that includes l. In the following we write $\mathit{rg}(f)$ to denote the *range* (or co-domain) of a function f.

Definition 10 (Locally-fenced action model)**.** An action model $\mathcal{I} \in \mathsf{AMod}$ is *locally fenced* by $\mathcal{F} \in \mathcal{P}(\mathsf{LState})$, written $\mathcal{F} \blacktriangleright \mathcal{I}$, iff for all $g \in \mathcal{F}$ and all $a \in \mathit{rg}(\mathcal{I})$,

$$g \copyright\, a \wedge (\mathit{potential}(a, g) \Rightarrow a[g] \in \mathcal{F})$$

Definition 11 (Action model confinement)**.** An action model $\mathcal{I}$ is *confined* to a logical state l, written $l \copyright\, \mathcal{I}$, if there exists a fence $\mathcal{F}$ such that $l \in \mathcal{F}$ and $\mathcal{F} \blacktriangleright \mathcal{I}$.

We are almost in a position to define well-formedness of worlds. Since capabilities enable the manipulation of the shared state through their associated actions in the action models, for a world $(l, g, \mathcal{I})$ to be well-formed the capabilities found in the local state l and shared state g must be *contained* in the action model $\mathcal{I}$. That is, *all* capabilities found in the combined state $l \circ g$ must be accounted for in $\mathcal{I}$.

Definition 12 (Capability containment)**.** A capability $\kappa \in \mathbb{K}$, is *contained* in an action model $\mathcal{I} \in \mathsf{AMod}$, written $\kappa \prec \mathcal{I}$ iff

$$\exists K \in \mathcal{P}(\mathbb{K}) \,.\, \kappa = \prod_{\kappa_i \in K}^{\bullet_{\mathbb{K}}} \kappa_i \wedge \forall \kappa_i \in K. \exists \kappa' \in \mathsf{dom}(\mathcal{I}) \,.\, \kappa_i \leq \kappa'$$

The above states that κ is contained in $\mathcal{I}$ iff κ is the composition of smaller capabilities $\kappa_i \in K$ where each constituent κ_i is accounted for in the domain of $\mathcal{I}$.

In order to ensure indefinite extension of the shared state and creation of fresh capabilities (through Extend principle), as part of the well-formedness condition we require that the domain of action models are well-defined. That is, one can always guarantee the existence of a capability that is fresh (disjoint from the domain of the action model); and that the domain remains well-defined after extension. This is captured by the following definition of well-defined capability sets.

Definition 13 (Well-defined capability subsets)**.** A capability set $K \in \mathcal{P}(\mathbb{K})$ is *well defined*, written $\diamond K$, if and only if there exists a set $\chi \in \mathcal{P}(\mathcal{P}(\mathbb{K}))$ such that $K \in \chi$ and

$$\begin{array}{l} \forall \kappa \in \mathbb{K}.\, \{\kappa\} \in \chi \\ \wedge\, \forall K' \in \chi.\, \exists \kappa \in \mathbb{K}.\, \kappa \perp K' \\ \wedge\, \forall K_1, K_2 \in \chi.\, K_1 \cup K_2 \in \chi \end{array}$$

where we write $\kappa \perp K$ to denote $\forall \kappa' \in K.\, \kappa \perp \kappa'$.

We can now formalise the notion of well-formedness. A world $(l, g, \mathcal{I})$ is well-formed if l and g are compatible, the capabilities found in $l \circ g$ are contained in the action model $\mathcal{I}$,$\mathcal{I}$ is confined to g, and the domain of $\mathcal{I}$ is well-defined.

Definition 14 (Well-formedness)**.** A 4-tuple $(l, g, \mathcal{I})$ is *well-formed*, written $\mathsf{wf}(l, g, \mathcal{I})$, iff

$$(\exists m, \kappa.\, l \circ g = (m, \kappa) \wedge \kappa \prec \mathcal{I}) \wedge g \circledcirc \mathcal{I} \wedge \diamond(\mathsf{dom}(\mathcal{I}))$$

Definition 15 (Worlds)**.** The set of *worlds* is defined as

$$\mathsf{World} \stackrel{\text{def}}{=} \{w \in \mathsf{LState} \times \mathsf{LState} \times \mathsf{AMod} \mid \mathsf{wf}(w)\}$$

The *composition* $w \bullet w'$ of two worlds $\bullet : \mathsf{World} \to \mathsf{World} \rightharpoonup \mathsf{World}$, is defined as follows.

$$(l, g, \mathcal{I}) \bullet (l', g', \mathcal{I}') \stackrel{\text{def}}{=} \begin{cases} (l \circ l', g, \mathcal{I}) & \text{if } g = g', \text{ and } \mathcal{I} = \mathcal{I}' \\ & \text{and } \mathsf{wf}((l \circ l', g, \mathcal{I})) \\ \textit{undefined} & \text{otherwise} \end{cases}$$

The set of worlds with composition $\bullet$ forms a separation algebra with multiple units: all well-formed states of the form $(\mathbf{0}_{\mathsf{L}}, g, \mathcal{I})$. Given a world w, we write w_{L} for the first projection.

1.2 Assertions

Our assertions extend standard assertions from separation logic with *subjective views* and *capability assertions*. We assume an infinite set, LVar, of *logical variables* and a set of *logical environments* $\iota \in \mathsf{LEnv} : \mathcal{P}(\mathsf{LVar} \to \mathsf{Val})$ that associate logical variables with their values.

CoLoSL is parametric with respect to the machine states and capability assertions and can be instantiated with any assertion language over machine states $\mathbb{M}$ and capabilities $\mathbb{K}$. This is captured by the following parameters.

Parameter 3 (Machine state assertions)**.** Assume a set of *machine state assertions* MAssn, ranged over by $\mathcal{M}, \mathcal{M}_1, \cdots, \mathcal{M}_n$ and an associated semantics function:

$$(\!|.|\!)^M_{(.)} : \mathsf{MAssn} \to \mathsf{LEnv} \to \mathcal{P}(\mathbb{M})$$

Parameter 4 (Capability assertions)**.** Assume a set of capability assertions KAssn ranged over by $\mathcal{K}, \mathcal{K}_1, \cdots, \mathcal{K}_n$ and an associated semantics function:

$$(\!|.|\!)^K_{(.)} : \mathsf{KAssn} \to \mathsf{LEnv} \to \mathcal{P}(\mathbb{K})$$

Definition 16 (Assertion syntax)**.** The assertions of CoLoSL are elements of Assn described by the grammar below, where x ranges over logical variables.

$$\begin{aligned}
A &::= \mathit{false} \mid \mathsf{emp} \mid \mathcal{M} \mid \mathcal{K} \\
\mathsf{LAssn} \ni p, q &::= A \mid \neg p \mid \exists x.\, p \mid p \vee q \mid p * q \mid p \uplus q \mid p -\!\!\circledast q \\
\mathsf{Assn} \ni P, Q &::= p \mid \exists x.\, P \mid P \vee Q \mid P * Q \mid P \uplus Q \mid \boxed{P}_I \\
\mathsf{IAssn} \ni I &::= \emptyset \mid \{\mathcal{K} : \exists \bar{y}.\, P \rightsquigarrow Q\} \cup I
\end{aligned}$$

This syntax follows from standard separation logic, with the exception of subjective views $\boxed{P}_I$. emp is true of the units of $\bullet$. Machine state assertions ($\mathcal{M}$) and capability assertions ($\mathcal{K}$) are interpreted over a world's local state: $\mathcal{M}$ is true of a local state $(m, \mathbf{0}_{\mathbb{K}})$ where m satisfies $\mathcal{M}$; similarly, $\mathcal{K}$ is true of a local state $(\mathbf{0}_{\mathbb{M}}, \kappa)$ where κ satisfies $\mathcal{K}$. $P * Q$ is true of worlds that can be split into two according to $\bullet$ such that one state satisfies P and the other satisfies Q; $P \uplus Q$ is the *overlapping conjunction*, true of worlds can be split three-way according to $\bullet$, such that the $\bullet$-composition of the first two worlds satisfies P and the $\bullet$-composition of the last two satisfy Q [16]; classical predicates and connectives have their standard classical meaning. Interference assertions I describe actions enabled by a given capability, in the form of a pre- and post-condition.

A subjective view $\boxed{P}_I$ is true of $(l, g, \mathfrak{I})$ when $l = \mathbf{0}_{\mathsf{L}}$ and a subjective view s can be found in the global shared state g, *i.e.* $g = s \circ r$ for some *context* r, such that s satisfies P in the standard separation logic sense, and I and $\mathfrak{I}$ *agree* given the decomposition s, r, in the following sense:

1. every action in I is reflected in $\mathfrak{I}$;
2. every action in $\mathfrak{I}$ that is potentially enabled in g and has a visible effect on s is reflected in I;
3. the above holds after any number of action applications in $\mathfrak{I}$ affecting g

These conditions will be captured by the *action model closure* relation $\mathfrak{I}{\downarrow}\,(s, r, \langle\!|I|\!\rangle_\iota)$ given by the upcoming Def. 24 (where $\langle\!|I|\!\rangle_\iota$ is the interpretation of I given a logical environment ι).

The semantics of CoLoSL assertions is given by a forcing relation $w, \iota \vDash P$ between a world w, a logical environment $\iota \in \mathsf{LEnv}$, and a formula P. We use two auxiliary forcing relations. The first one $l, \iota \vDash_{\mathsf{SL}} P$ interprets formulas P in the usual separation logic sense over a logical state l (and ignores shared state assertions). The second one $s, \iota \vDash_{g,\mathfrak{I}} P$ interprets assertions over a *subjective view* s that is part of the global shared state g, subject to action model $\mathfrak{I}$. This third form of satisfaction is needed to deal with nesting of subjective views. We often write $\vDash_\dagger$ as a shorthand for $\vDash_{g,\mathfrak{I}}$ when we do not need to refer to the individual components g and $\mathfrak{I}$.

Note that this presentation with several forcing relations differs from the usual CAP presentation [6], where formulas are first interpreted over worlds that are not necessarily well-formed, and then cut down to well-formed ones. The CAP presentation strays from separation logic models in some respects; for instance, in CAP, $*$ is not the adjoint of $-\!\!*$, the "magic wand" connective of separation logic. Although we have omitted this connective from our presentation, its definition in CoLoSL would be standard and satisfy the adjunction with $*$.

Definition 17 (Assertion semantics)**.** Given a logical environment $\iota \in \mathsf{LEnv}$, the semantics of CoLoSL assertions is as follows, where $\langle\!|.|\!\rangle_{(.)} : \mathsf{IAssn} \to \mathsf{LEnv} \to \mathsf{AMod}$ denotes the semantics of interference assertions and $\mathfrak{I}\downarrow (s, r, \langle\!|I|\!\rangle_\iota)$ will be given in Def. 24.

$$\begin{array}{lll}
(l, g, \mathfrak{I}), \iota \vDash p & \text{iff} & l, \iota \vDash_{\mathsf{SL}} p \\
(l, g, \mathfrak{I}), \iota \vDash \boxed{P}_I & \text{iff} & l = \mathbf{0}_{\mathbb{M}} \text{ and } \exists s, r.\, g = s \circ r \text{ and} \\
 & & s, \iota \vDash_{g,\mathfrak{I}} P \text{ and } \mathfrak{I}{\downarrow}\,(s, r, \langle\!|I|\!\rangle_\iota)
\end{array}$$

$$
\begin{array}{lll}
w, \iota \vDash \exists x.\, P & \text{iff} & \exists v.\, w, [\iota \mid x : v] \vDash P \\
w, \iota \vDash P \vee Q & \text{iff} & w, \iota \vDash P \text{ or } w, \iota \vDash Q \\
w, \iota \vDash P_1 \ast P_2 & \text{iff} & \exists w_1, w_2.\, w = w_1 \bullet w_2 \text{ and} \\
 & & w_1, \iota \vDash P_1 \text{ and } w_2, \iota \vDash P_2 \\
w, \iota \vDash P_1 \mathbin{\cup\!\!\!\!\ast} P_2 & \text{iff} & \exists w', w_1, w_2.\, w = w' \bullet w_1 \bullet w_2 \text{ and} \\
 & & w' \bullet w_1, \iota \vDash P_1 \text{ and } w' \bullet w_2, \iota \vDash P_2
\end{array}
$$

where

$$
\begin{array}{lll}
s, \iota \vDash_{g,\mathcal{I}} p & \text{iff} & s, \iota \vDash_{\mathsf{SL}} p \\
s, \iota \vDash_{g,\mathcal{I}} \boxed{P}_I & \text{iff} & (s, g, \mathcal{I}), \iota \vDash \boxed{P}_I \\
s, \iota \vDash_{\dagger} \exists x.\, P & \text{iff} & \exists v.\, s, [\iota \mid x : v] \vDash_{\dagger} P \\
s, \iota \vDash_{\dagger} P \vee Q & \text{iff} & s, \iota \vDash_{\dagger} P \text{ or } s, \iota \vDash_{\dagger} Q \\
s, \iota \vDash_{\dagger} P_1 \ast P_2 & \text{iff} & \exists s_1, s_2.\, s = s_1 \circ s_2 \text{ and} \\
 & & s_1, \iota \vDash_{\dagger} P_1 \text{ and } s_2, \iota \vDash_{\dagger} P_2 \\
s, \iota \vDash_{\dagger} P_1 \mathbin{\cup\!\!\!\!\ast} P_2 & \text{iff} & \exists s', s_1, s_2.\, s = s' \circ s_1 \circ s_2 \text{ and} \\
 & & s' \circ s_1, \iota \vDash_{\dagger} P_1 \text{ and } s' \circ s_2, \iota \vDash_{\dagger} P_2
\end{array}
$$

$$
\begin{array}{lll}
l, \iota \vDash_{\mathsf{SL}} \mathit{false} & & \text{never} \\
l, \iota \vDash_{\mathsf{SL}} \mathsf{emp} & \text{iff} & l = \mathbf{0}_{\mathsf{L}} \\
l, \iota \vDash_{\mathsf{SL}} \mathcal{M} & \text{iff} & \exists m.\, l = (m, \mathbf{0}_{\mathbb{K}}) \text{ and } m \in \llparenthesis \mathcal{M} \rrparenthesis_{\iota}^{M} \\
l, \iota \vDash_{\mathsf{SL}} \mathcal{K} & \text{iff} & \exists \kappa.\, l = (\mathbf{0}_{\mathbb{M}}, \kappa) \text{ and } \kappa \in \llparenthesis \mathcal{K} \rrparenthesis_{\iota}^{K} \\
l, \iota \vDash_{\mathsf{SL}} \neg p & \text{iff} & l, \iota \nvDash_{\mathsf{SL}} p \\
l, \iota \vDash_{\mathsf{SL}} p \mathrel{-\!\!\circledast} q & \text{iff} & \exists l'.\, l', \iota \vDash_{\mathsf{SL}} p \text{ and } l \sharp l' \\
 & & \text{ implies } l \circ l', \iota \vDash_{\mathsf{SL}} q \\
l, \iota \vDash_{\mathsf{SL}} P_1 \ast P_2 & \text{iff} & \exists l_1, l_2.\, l = l_1 \circ l_2 \text{ and} \\
 & & l_1, \iota \vDash_{\mathsf{SL}} P_1 \text{ and } l_2, \iota \vDash_{\mathsf{SL}} P_2 \\
l, \iota \vDash_{\mathsf{SL}} P \vee Q & \text{iff} & l, \iota \vDash_{\mathsf{SL}} P \text{ or } l, \iota \vDash_{\mathsf{SL}} Q \\
l, \iota \vDash_{\mathsf{SL}} \exists x.\, P & \text{iff} & \exists v.\, l, [\iota \mid x : v] \vDash_{\mathsf{SL}} P \\
l, \iota \vDash_{\mathsf{SL}} P_1 \mathbin{\cup\!\!\!\!\ast} P_2 & \text{iff} & \exists l', l_1, l_2.\, l = l' \circ l_1 \circ l_2 \text{ and} \\
 & & l' \circ l_1, \iota \vDash_{\mathsf{SL}} P_1 \text{ and } l' \circ l_2, \iota \vDash_{\mathsf{SL}} P_2
\end{array}
$$

and

$$\langle\!|I|\!\rangle_\iota(\kappa) \overset{\text{def}}{=} \left\{ (p, q, c \circ l) \;\middle|\; \begin{array}{l} \mathcal{K} : \exists\vec{y}.\, P \rightsquigarrow Q \in I \wedge \kappa \in (\!|\mathcal{K}|\!)_\iota^K \wedge \\ \exists\vec{v}, \mathcal{I}. \\ \quad p \circ c, [\iota \mid \vec{y} : \vec{v}] \vDash_{(p \circ c \circ r \circ l), \mathcal{I}} P \wedge \\ \quad q \circ c, [\iota \mid \vec{y} : \vec{v}] \vDash_{(q \circ c \circ r \circ l), \mathcal{I}} Q \wedge \\ \quad \forall l'.\, l' \leq p \wedge l' \leq q \implies l' = \mathbf{0}_\mathsf{L} \end{array} \right\}$$

Given a logical environment ι, we write $[\![P]\!]_\iota$ for the set of all worlds satisfying P. That is,

$$[\![P]\!]_\iota \overset{\text{def}}{=} \{w \mid w, \iota \vDash P\}$$

Although CoLoSL allows for nested subjective views (e.g. $\boxed{P * \boxed{Q}_{I'}}_I$), it is often useful to *flatten* them into equivalent assertions with no nested boxes. We introduce *flat assertions*, $\mathsf{FAssn} \subset \mathsf{Assn}$, to capture a subset of assertions with no nested subjected views and provide a *flattening mechanism* to rewrite CoLoSL assertions as equivalent flat assertions.

Definition 18 (Flattening)**.** The set of *flat assertions* FAssn is defined by the following grammar.

$$\mathsf{FAssn} \ni P, Q ::= p \mid \boxed{p}_I \mid \exists x.\, P \mid P \vee Q \mid P * Q \mid P \uplus\!\!\!* \; Q$$

The *flattening* function $\mathsf{f}(.) : \mathsf{Assn} \to \mathsf{FAssn}$ is defined inductively as follows where x does not appear free in I and $\odot \in \{\vee, *, \uplus\!\!\!*\}$.

$$\begin{array}{ll} \mathsf{f}(p) \overset{\text{def}}{=} p & \mathsf{f}(P \odot Q) \overset{\text{def}}{=} \mathsf{f}(P) \odot \mathsf{f}(Q) \\ \mathsf{f}(\exists x.\, P) \overset{\text{def}}{=} \exists x.\, \mathsf{f}(P) & \mathsf{f}\left(\boxed{\boxed{P}_I * Q}_{I'}\right) \overset{\text{def}}{=} \mathsf{f}\left(\boxed{P}_I\right) * \mathsf{f}\left(\boxed{Q}_{I'}\right) \\ \mathsf{f}\left(\boxed{p}_I\right) \overset{\text{def}}{=} \boxed{p}_I & \mathsf{f}\left(\boxed{\boxed{P}_I \uplus\!\!\!* \; Q}_{I'}\right) \overset{\text{def}}{=} \mathsf{f}\left(\boxed{P}_I\right) * \mathsf{f}\left(\boxed{Q}_{I'}\right) \\ \mathsf{f}\left(\boxed{\exists x.\, P}_I\right) \overset{\text{def}}{=} \exists x.\, \mathsf{f}\left(\boxed{P}_I\right) & \mathsf{f}\left(\boxed{P \vee Q}_I\right) \overset{\text{def}}{=} \mathsf{f}\left(\boxed{P}_I\right) \vee \mathsf{f}\left(\boxed{Q}_I\right) \end{array}$$

Definition 19 (Erasure)**.** The *erasure* of an assertion $\lfloor(.) : \mathsf{Assn} \to \mathsf{LAssn}$ is defined inductively over the structure of assertions as follows with $\odot \in \{\vee, *, \uplus\!\!\!*\}$

$$\lfloor A \overset{\text{def}}{=} A \qquad \lfloor\left(\boxed{P}_I\right) \overset{\text{def}}{=} \mathsf{emp} \qquad \lfloor(\exists x.\, P) \overset{\text{def}}{=} \exists x.\ \lfloor P \qquad \lfloor(P \odot Q) \overset{\text{def}}{=} \lfloor P \odot \lfloor Q$$

Definition 20 (Gather/merge)**.** Given an assertion $P \overset{\text{def}}{=} \exists\bar{x}.\, P' \in \mathsf{Assn}$, where P is in the prenex normal form with no bound variables in P',the *gather*, $\bigotimes(.) : \mathsf{Assn} \to \mathsf{LAssn}$, and *merge*, $\bigoplus(.) : \mathsf{Assn} \to \mathsf{LAssn}$, operations are defined as follows:

$$\bigotimes P \overset{\text{def}}{=} \exists \bar{x}.\, p \ast q \qquad \bigoplus P \overset{\text{def}}{=} \exists \bar{x}.\, p \uplus q \qquad \text{where } (p, q) = \mathsf{ub}\left(\mathsf{f}\left(P'\right)\right)$$

with the auxiliary function $\mathsf{ub}\,(.) : \mathsf{FAssn} \to (\mathsf{LAssn} \times \mathsf{LAssn})$ defined inductively over the structure of local assertions as follows where $\odot \in \{\ast, \uplus\}$. In what follows, $\mathsf{ub}\,(P) = (p, p')$ and $\mathsf{ub}\,(Q) = (q, q')$ where applicable.

$$\mathsf{ub}\,(p) \overset{\text{def}}{=} (p, \mathsf{emp}) \qquad \mathsf{ub}\left(\boxed{p}_{\,I}\right) \overset{\text{def}}{=} (\mathsf{emp}, p) \qquad \mathsf{ub}\,(P \odot Q) \overset{\text{def}}{=} \left(p \odot q, p' \uplus q'\right)$$

$$\mathsf{ub}\,(P \vee Q) \overset{\text{def}}{=} \left(p \vee q, p' \vee q'\right)$$

Definition 21 (Interference entailment)**.** An interference assertion I *entails* interference assertion I', written $I \vdash_{\mathrm{I}} I'$, if

$$\forall \iota \in \mathsf{LEnv}.\ \langle\!| I |\!\rangle_\iota \subseteq \left\langle\!\left| I' \right|\!\right\rangle_\iota$$

Lemma 1. The following judgements are valid.

$$\frac{}{P \Leftrightarrow \mathsf{f}\,(P)} \qquad \frac{}{P \vdash \downharpoonleft P} \qquad \frac{P \vdash Q}{\boxed{P}_{\,I} \vdash \boxed{Q}_{\,I}} \qquad \frac{P \uplus Q \vdash \mathit{false}}{\boxed{P}_{\,I} \ast \boxed{Q}_{\,I'} \vdash \mathit{false}}$$

$$\frac{R \vdash P \quad P \uplus Q \vdash R \uplus Q}{\boxed{P}_{\,I} \ast \boxed{Q}_{\,I'} \vdash \boxed{R}_{\,I} \ast \boxed{Q}_{\,I'}} \qquad \frac{I_1 \vdash_{\mathrm{I}} I_2}{I \cup I_1 \vdash_{\mathrm{I}} I \cup I_2}$$

$$\frac{}{\{\mathcal{K} : \exists \bar{y}.\, P \rightsquigarrow Q\} \vdash_{\mathrm{I}} \{\mathcal{K} : \exists \bar{y}.\, \bigoplus P \rightsquigarrow \bigoplus Q\}}$$

$$\frac{P \vdash P' \quad Q \vdash Q'}{\{\mathcal{K} : \exists \bar{y}.\, P \rightsquigarrow Q\} \vdash_{\mathrm{I}} \{\mathcal{K} : \exists \bar{y}.\, P' \rightsquigarrow Q'\}}$$

Action Model Closure Let us now turn to the definition of action model closure, as informally introduced at the beginning of this section. First, we need to revisit the effect of actions to take into account the splitting of the global shared state into a subjective state s and a context r.

Definition 22 (Action application (cont.))**.** The *application* of action $a = (p, q, c)$ on the subjective state s together with the context r, written $a[s, r]$, is defined provided that $a[s \circ r]$ is defined. When that is the case, we write $a[s, r]$ for

$$\begin{aligned} &\left\{ (q \circ s', r') \;\middle|\; \begin{array}{l} p = p_s \circ p_r \wedge p_s > \mathbf{0}_{\mathsf{L}} \\ \wedge\ s = p_s \circ s' \wedge r = p_r \circ r' \end{array} \right\} \\ \cup\ &\left\{ (s, q \circ r') \mid r = p \circ r' \right\} \end{aligned}$$

We observe that this new definition and Def. 8 are linked in the following way:

$$\forall s', r' \in a[s, r].\, s' \circ r' = a[s \circ r]$$

In our informal description of action model closure in Def. 16 we stated that some actions must be *reflected* in some action models. Intuitively, an action is reflected in an action model if for every state in which the action can take place, the action model includes an action with a similar effect that can also occur in that state. In other words, $a = (p, q, c)$ is reflected in $\mathcal{I}$ from a state l if whenever a is enabled in l (i.e. $p \circ c \le l$), then there exists an action $a' = (p, q, c') \in \mathcal{I}$ (with the same pre- and post states p and q) that is also enabled in l (i.e. $p \circ c' \le l$). We proceed with the definition of action reflection.

Definition 23. An action $a = (p, q, c)$ is *reflected* in a set of actions A from a state l, written *reflected*(a, l, A), if

$$\forall r.\, p \circ c \le l \circ r \Rightarrow \exists c'.\, (p, q, c') \in A \wedge p \circ c' \le l \circ r$$

Let us now give the formal definition of action model closure. For each condition mentioned in Def. 16, we annotate which part of the definition implements them. Given an action $a = (p, q, c)$, we write

$$\mathit{enabled}(a, g) \overset{\text{def}}{=} \mathit{potential}(a, g) \wedge p \circ c \le g$$

That is, a can actually happen in g since g holds all the resources in both the pre-state of a and its condition.

Definition 24 (Action model closure)**.** An action model $\mathcal{I}$ is *closed* under a subjective state s, context r, and action model $\mathcal{I}'$, written $\mathcal{I}{\downarrow} (s, r, \mathcal{I}')$, if $\mathcal{I}' \subseteq \mathcal{I}$ and $\mathcal{I} {\downarrow_n} (s, r, \mathcal{I}')$ for all $n \in \mathbb{N}$, where $\downarrow_n$ is defined recursively as follows:

$$\begin{array}{ll}
\mathcal{I}{\downarrow_0} (s, r, \mathcal{I}') \overset{\text{def}}{\iff} \mathit{true} & \\
\mathcal{I}{\downarrow_{n+1}} (s, r, \mathcal{I}') \overset{\text{def}}{\iff} & \\
\quad (\forall \kappa.\, \forall a \in \mathcal{I}(\kappa).\, \mathit{potential}(a, s \circ r) \Rightarrow & \\
\qquad (\mathit{reflected}(a, s \circ r, \mathcal{I}'(\kappa)) \vee \neg \mathit{visible}(a, s)) & (2) \\
\qquad \wedge\, \forall (s', r') \in a[s, r].\, \mathcal{I}{\downarrow_n} (s', r', \mathcal{I}') & (3)
\end{array}$$

Recall from the semantics of the assertions (Def. 17) that $\mathcal{I}'$ corresponds to the interpretation of an interference assertion I in a subjective view. The first condition($\mathcal{I}' \subseteq \mathcal{I}$) asserts that the subjective action model $\mathcal{I}'$ is contained in $\mathcal{I}$ (*cf.* property (1) in Def. 16); consequently (from the semantics of subjective assertions), $\mathcal{I}$ represents the superset of all interferences known to subjective views.

The above states that the $\mathcal{I}$ is closed under $(s, r, \mathcal{I}')$ if the closure relation holds for any number of steps $n \in \mathbb{N}$ where

- s denotes the subjective view of the shared state;
- r denotes the context;
- $s \circ r$ captures the entire shared state;
- a step corresponds to the occurrence of an action as prescribed in $\mathcal{I}$ which may or may not be found in $\mathcal{I}'$ (since $\mathcal{I} \subseteq \mathcal{I}$).

The relation is satisfied trivially for no steps ($n = 0$). On the other hand, for an arbitrary $n \in \mathbb{N}$ the relation holds iff for any action a in $\mathcal{I}$, where a is potentially enabled in $s \circ r$, then *either*:

a. a is reflected in $\mathcal{I}'$ (a is known to the subjective view), and $\mathcal{I}$ is also closed under any subjective state s' and context r' resulting from application of a ($(s', r') \in a[s, r]$); *or*

b. a does not affect the subjective state s (a is not visible in s), and $\mathcal{I}$ is also closed under any subjective state s' and context r' resulting from application of a ($(s', r') \in a[s, r]$).

Note that in the last item above (b), since a is not visible in s, given any $(s', r') \in a[s, r]$, from the definition of action application we then know $s' = s$.

We make further observations about this definition. First, property (3) makes our assertions robust with respect to future extensions of the shared state, where potentially enabled actions may be become enabled using additional catalyst that is not immediately present. Second, $\mathcal{I}'$ (and thus interference assertions) need not reflect actions that have no visible effect on the subjective state.

This completes the definition of the semantics of assertions. We can now show that the logical principles of CoLoSL are sound. The proof of the Shift and Extend principles will be delayed until the following chapter, where $\Rrightarrow$ is defined.

Lemma 2. The logical implications Copy, Forget, and Merge are *valid*; *i.e.* true of all worlds and logical interpretations.

Proof. The case of Copy is straightforward from the semantics of assertions. In order to show that the Forget implication is valid, it suffices to show:

$$\forall \iota \in \mathsf{LEnv}.\ \left\{ w \mid w, \iota \vDash \boxed{P \mathbin{\cup\!\!\!\!\ast} Q}_I \right\} \subseteq \left\{ w \mid w, \iota \vDash \boxed{P}_I \right\}$$

We first demonstrate that, whenever an action model is closed under a subjective state $s_1 \circ s_2$ and context r, then it is also closed under the smaller subjective state s_1, and the larger context extended with the forgotten state. That is,

$$\begin{array}{l} \forall s_1, s_2, r \in \mathsf{LState}.\ \forall \mathcal{J}, \mathcal{J}' \in \mathsf{AMod}. \\ \qquad \mathcal{J}{\downarrow} (s_1 \circ s_2, r, \mathcal{J}') \implies \mathcal{J}{\downarrow} (s_1, s_2 \circ r, \mathcal{J}') \end{array}$$

This is formalised in Lemma 7 of Appendix A. We then proceed to establish the desired result by calculation:

$$\begin{array}{rl}
\left\{ w \mid w, \iota \vDash \boxed{P \mathbin{\cup\!\!\!\!\ast} Q}_I \right\} & = \left\{ \left(\begin{array}{l} (\mathbf{0}_{\mathsf{L}}, \\ (s \circ r), \mathcal{J} \end{array} \right) \middle| \begin{array}{l} \exists s_p, s_c, s_q.\ s = s_p \circ s_c \circ s_q \\ \wedge r \in \mathsf{LState} \\ \wedge (s_p \circ s_c), \iota \vDash_{(s \circ r), \mathcal{J}} P \\ \wedge (s_q \circ s_c), \iota \vDash_{(s \circ r), \mathcal{J}} Q \\ \wedge \mathcal{J}{\downarrow} (s_p \circ s_c \circ s_q,\ r, \llparenthesis I \rrparenthesis_\iota) \end{array} \right\} \\
\text{By Lemma 7} & \subseteq \left\{ \left(\begin{array}{l} (\mathbf{0}_{\mathsf{L}}, \\ (s \circ r), \mathcal{J} \end{array} \right) \middle| \begin{array}{l} \exists s_p, s_c, s_q.\ s = s_p \circ s_c \circ s_q \\ \wedge r \in \mathsf{LState} \\ \wedge (s_p \circ s_c), \iota \vDash_{(s \circ r), \mathcal{J}} P \\ \wedge (s_q \circ s_c), \iota \vDash_{(s \circ r), \mathcal{J}} Q \\ \wedge \mathcal{J}{\downarrow} (s_p \circ s_c,\ s_q \circ r, \llparenthesis I \rrparenthesis_\iota) \end{array} \right\} \\
& \subseteq \left\{ \left(\begin{array}{l} (\mathbf{0}_{\mathsf{L}}, \\ (s_p \circ r), \mathcal{J} \end{array} \right) \middle| \begin{array}{l} s_p, \iota \vDash_{(s_p \circ r), \mathcal{J}} P \wedge r \in \mathsf{LState} \\ \wedge \mathcal{J}{\downarrow} (s_p,\ r, \llparenthesis I \rrparenthesis_\iota) \end{array} \right\} \\
& = \left\{ w \mid w, \iota \vDash \boxed{P}_I \right\}
\end{array}$$

as required.

Similarly, to show that the Merge implication is valid, it suffices to show:

$$\forall \iota \in \mathsf{LEnv}.\ \left\{ w \mid w, \iota \vDash \boxed{P}_{I_1} \ast \boxed{Q}_{I_2} \right\} \subseteq \left\{ w \mid w, \iota \vDash \boxed{P \mathbin{\cup\!\!\!\!\ast} Q}_{I_1 \cup I_2} \right\}$$

We first demonstrate that, whenever an action model closure relation holds for two (potentially) different subjective states (that may overlap), subject to two (potentially) different interference relations, then the closure relation also holds for the combined subjective states and their associated interference relations. That is:

$$\begin{array}{l} \forall s_p, s_q, s_c, r \in \mathsf{LState}.\ \forall \mathcal{J}, \mathcal{J}_1, \mathcal{J}_2 \in \mathsf{AMod}. \\ \quad \mathcal{J}{\downarrow} (s_p \circ s_c, s_q \circ r, \mathcal{J}_1) \ \wedge\ \mathcal{J}{\downarrow} (s_q \circ s_c, s_p \circ r, \mathcal{J}_2) \implies \\ \qquad\qquad \mathcal{J}{\downarrow} (s_p \circ s_c \circ s_q, r, \mathcal{J}_1 \cup \mathcal{J}_2) \end{array}$$

This is formalised in Lemma 8 of Appendix A. We then proceed to establish the desired result by calculation:

$$
\begin{array}{rl}
& \left\{ w \mid w, \iota \vDash \boxed{P}_{I_1} \star \boxed{Q}_{I_2} \right\} \\
= & \left\{ (\mathbf{0}_{\mathsf{L}}, s \circ r, \mathfrak{I}) \;\middle|\; \begin{array}{l} \exists s_p, s_c, s_q.\, s = s_p \circ s_c \circ s_q \\ \wedge r \in \mathsf{LState} \\ \wedge (s_p \circ s_c), \iota \vDash_{(s \circ r), \mathfrak{I}} P \\ \wedge (s_q \circ s_c), \iota \vDash_{(s \circ r), \mathfrak{I}} Q \\ \wedge \mathfrak{I}{\downarrow} \left(s_p \circ s_c,\ s_q \circ r, \llparenthesis I_1 \rrparenthesis_{\iota}\right) \\ \wedge \mathfrak{I}{\downarrow} \left(s_c \circ s_q,\ s_p \circ r, \llparenthesis I_2 \rrparenthesis_{\iota}\right) \end{array} \right\} \\
\text{By Lemma 8} \subseteq & \left\{ (\mathbf{0}_{\mathsf{L}}, s \circ r, \mathfrak{I}) \;\middle|\; \begin{array}{l} \exists s_p, s_c, s_q.\, s = s_p \circ s_c \circ s_q \\ \wedge r \in \mathsf{LState} \\ \wedge (s_p \circ s_c), \iota \vDash_{(s \circ r), \mathfrak{I}} P \\ \wedge (s_q \circ s_c), \iota \vDash_{(s \circ r), \mathfrak{I}} Q \\ \wedge \mathfrak{I}{\downarrow} \left(s_p \circ s_c \circ s_q, r, \llparenthesis I_1 \cup I_2 \rrparenthesis_{\iota}\right) \end{array} \right\} \\
\subseteq & \left\{ (\mathbf{0}_{\mathsf{L}}, s \circ r, \mathfrak{I}) \;\middle|\; \begin{array}{l} s, \iota \vDash_{(s \circ r), \mathfrak{I}} P \mathbin{\cup\!\!\!\!\ast} Q \\ \wedge r \in \mathsf{LState} \\ \wedge \mathfrak{I}{\downarrow} \left(s, r, \llparenthesis I_1 \cup I_2 \rrparenthesis_{\iota}\right) \end{array} \right\} \\
= & \left\{ w \mid w, \iota \vDash \boxed{P \mathbin{\cup\!\!\!\!\ast} Q}_{I_1 \cup I_2} \right\}
\end{array}
$$

□

Note that, the version of FORGET where predicates are conjoined using $\mathbin{\cup\!\!\!\!\ast}$ is also valid for all P, Q, and I, as shown by the following derivation, where the first implication follows from the properties of $\mathbin{\cup\!\!\!\!\ast}$.

$$\boxed{P \mathbin{\cup\!\!\!\!\ast} Q}_{I} \overset{\text{WEAKEN}}{\Rightarrow} \boxed{P \star \mathit{true}}_{I} \overset{\text{FORGET}}{\Rightarrow} \boxed{P}_{I}$$

2. CoLoSL Program Logic

This section introduces the core concepts behind our program logic. We start by defining what the rely and guarantee conditions of each thread are in terms of their action models. This allows us to define *stability* against rely conditions, *repartitioning*, which logically represents a thread's atomic actions (and have to be in the guarantee condition), and *semantic implication*. Equipped with these notions, we can justify the Shift and Extend principles.

2.1 Environment Semantics

Rely The rely relation represents potential interferences from the environment. Although the rely will be different for every program (and indeed, every thread), it is always defined in the same way, which we can break down into three kinds of possible interferences. In the remainder of this section, given a logical state $l = (m, \kappa)$, we write l_{M}, l_{K} for $fst(l)$ and $snd(l)$, respectively.

The first relation, R^{e}, extends the shared state g with new state g', along with a new interference $\mathcal{I}'$ on g'. We proceed with the definition of *action model extension* that captures the extension of the action model in such a way that respects all previous subjective views of a world; that is, the action model closure relation is preserved.

Definition 25 (Action model extension)**.** An action model $\mathcal{I}'$ *extends* $(g, \mathcal{I})$ with (g'), written $\mathcal{I}' \uparrow^{(g')} (g, \mathcal{I})$, iff for all $\mathcal{I}_0 \subseteq \mathcal{I}$ and s, r such that $s \circ r = g$:

$$\mathcal{I}\downarrow (s, r, \mathcal{I}_0) \implies \mathcal{I} \cup \mathcal{I}' \downarrow \left(s, r \circ g', \mathcal{I}_0\right)$$

Then the extension rely, R^{e}, is defined as

$$R^{\mathsf{e}} \stackrel{\text{def}}{=} \left\{ \begin{pmatrix} (l, g, \mathcal{I}), \\ (l, g \circ g', \mathcal{I} \cup \mathcal{I}') \end{pmatrix} \middle| \begin{array}{l} g'_{\mathsf{K}} \prec \mathsf{dom}\,(\mathcal{I}') \cup \mathsf{dom}\,(\mathcal{I}) \wedge \\ g' \copyright\, \mathcal{I}' \wedge \mathcal{I}' \uparrow^{(g')} (g, \mathcal{I}) \end{array} \right\}$$

The second kind of interference is the *update* of the global state according to actions in the global action model whose capability is "owned by the environment", *i.e.*, nowhere to be found in the current local and shared states.

$$R^{\mathsf{u}} \overset{\text{def}}{=} \left\{ \left((l, g, \mathcal{I}),\ (l, g', \mathcal{I}) \right) \middle| \exists \kappa.\ (l_{\mathsf{K}} \bullet_{\mathbb{K}} g_{\mathsf{K}}) \sharp \kappa \land (g, g') \in \lceil \mathcal{I} \rceil (\kappa) \right\}$$

where

$$\lceil \mathcal{I} \rceil (\kappa) \overset{\text{def}}{=} \{ (p \circ r, q \circ r) \mid (p, q) \in \mathcal{I}(\kappa) \land r \in \mathsf{LState} \}$$

The third and last kind of interference is the *shifting* of the local interference relation to a new one that allows more actions while preserving all current subjective views (as expressed by the global action model).

$$R^{\mathsf{s}} \overset{\text{def}}{=} \left\{ \left((l, g, \mathcal{I}), (l, g, \mathcal{I} \cup \mathcal{I}') \right) \;\middle|\; \mathcal{I}' \uparrow^{(\mathbf{0}_{\mathsf{L}})} (g, \mathcal{I}) \right\}$$

Definition 26 (Rely)**.** The *rely* relation $R : \mathcal{P}(\mathsf{World} \times \mathsf{World})$ is defined as follows, where $(\cdot)^*$ denotes the reflexive transitive closure:

$$R \overset{\text{def}}{=} \left(R^{\mathsf{u}} \cup R^{\mathsf{e}} \cup R^{\mathsf{s}} \right)^*$$

The rely relation enables us to define the stability of assertions with respect to the environment actions.

Definition 27 (Stability)**.** An assertion P is *stable* ($\mathsf{Stable}(P)$) if, for all $\iota \in \mathsf{LEnv}$ and $w, w' \in \mathsf{World}$, if $w, \iota \vDash P$ and $(w, w') \in R$, then $w', \iota \vDash P$.

Proving that an assertion is stable is not always obvious, in particular when there are numerous transitions to consider (all those in R^{e}, R^{u}, R^{s}); as it turns out, we only need to check stability against update actions in R^{u}, as expressed by the following lemma.

Lemma 3 (Stability)**.** If an assertion P is stable with respect to actions in R^{u}, then it is stable. That is, for all $w, w' \in \mathsf{World}$, $\iota \in \mathsf{LEnv}$, and $P \in \mathsf{Assn}$,

$$\text{if } w, \iota \vDash P \land (w, w') \in R \land \left(\forall w_1, w_2.\ \begin{pmatrix} w_1, \iota \vDash P \land \\ (w_1, w_2) \in R^{\mathsf{u}} \end{pmatrix} \Longrightarrow w_2, \iota \vDash P \right)$$

then $w', \iota \vDash P$

Proof. Given any arbitrary $n \in \mathbb{N}$, let $S^n \in \mathsf{World} \times \mathsf{World}$ denote a binary relation on worlds defined inductively as follows.

$$S^0 \overset{\text{def}}{=} R^{\mathsf{e}} \cup R^{\mathsf{u}} \cup R^{\mathsf{s}} \cup \{ (w, w) \mid w \in \mathsf{World} \}$$

$$S^{n+1} \stackrel{\text{def}}{=} \left\{(w, w') \mid (w, w'') \in S^0 \wedge (w'', w) \in S^n\right\}$$

Let $S \in \mathsf{World} \times \mathsf{World}$ denote a binary relation on worlds defined as follows.

$$S \stackrel{\text{def}}{=} \bigcup_{i \in \mathbb{N}} S^i$$

From the definition of the rely relation on worlds (R) we then have $R = S$. It thus suffices to show that for all $n \in \mathbb{N}$, $w, w' \in \mathsf{World}$, $\iota \in \mathsf{LEnv}$ and $P \in \mathsf{Assn}$,

$$\text{if } w, \iota \vDash P \wedge (w, w') \in S^n \wedge \left(\forall w_1, w_2.\ \begin{pmatrix} w_1, \iota \vDash P \wedge \\ (w_1, w_2) \in R^{\mathsf{u}} \end{pmatrix} \Longrightarrow w_2, \iota \vDash P\right)$$
$$\text{then } w', \iota \vDash P$$

We proceed by induction on n.

Base case $n = 0$
Pick an arbitrary $w, w' \in \mathsf{World}$, $\iota \in \mathsf{LEnv}$ and $P \in \mathsf{Assn}$ such that

$$(w, w') \in S^0 \tag{2.1}$$
$$w, \iota \vDash P \wedge \left(\forall w_1, w_2.\ \left(w_1, \iota \vDash P \wedge (w_1, w_2) \in R^{\mathsf{u}}\right) \Longrightarrow w_2, \iota \vDash P\right) \tag{2.2}$$

We are then required to show $w', \iota \vDash P$. From (2.1) there are then four cases to consider:

1. If $(w, w') \in \{(w, w) \mid w \in \mathsf{World}\}$, then $w' = w$ and from (2.1) we trivially have $w', \iota \vDash P$ as required.

2. If $(w, w') \in R^{\mathsf{e}}$ then from (2.2) and Lemma 22 we have $w', \iota \vDash P$ as required.

3. If $(w, w') \in R^{\mathsf{u}}$ then from (2.2) we trivially have $w', \iota \vDash P$ as required.

4. If $(w, w') \in R^{\mathsf{s}}$ then from (2.2) and Lemma 24 we have $w', \iota \vDash P$ as required.

Inductive case
Pick an arbitrary $w, w' \in \mathsf{World}$, $\iota \in \mathsf{LEnv}$ and $P \in \mathsf{Assn}$ such that

$$(w, w') \in S^{n+1} \tag{2.3}$$
$$w, \iota \vDash P \wedge \left(\forall w_1, w_2.\ \begin{pmatrix} w_1, \iota \vDash P \wedge \\ (w_1, w_2) \in R^{\mathsf{u}} \end{pmatrix} \Longrightarrow w_2, \iota \vDash P\right) \tag{2.4}$$

We are then required to show $w', \iota \vDash P$ provided that

$$\begin{array}{l}\forall m \leq n. \forall w_3, w_4 \in \mathsf{World}. \\ \quad \text{if } w_3, \iota \vDash P \land (w_3, w_4) \in S^m \land \left(\forall w_1, w_2. \begin{pmatrix} w_1, \iota \vDash P \land \\ (w_1, w_2) \in R^{\mathsf{u}} \end{pmatrix} \implies w_2, \iota \vDash P\right) \\ \quad \text{then } w_4, \iota \vDash P \end{array} \tag{I.H.}$$

From (2.3) and the definition of S^{n+1} we know there exists $w'' \in \mathsf{World}$ such that

$$(w, w'') \in S^0 \tag{2.5}$$
$$(w'', w') \in S^n \tag{2.6}$$

From (2.5), (2.4) and the base case we know

$$w'', \iota \vDash P \tag{2.7}$$

Consequently, from (2.4), (2.6), (2.7) and (I.H.) we have

$$w', \iota \vDash P$$

as required. □

We provide a number of syntactic judgements in Fig. 2.1 that allows us to ascertain the stability of an assertion without performing the necessary semantic checks. These judgements reduce stability checks to logical entailments. We appeal to the following auxiliary definition in our stability judgements.

Definition 28 (Combination)**.** The *combination* of an assertion $P \overset{\text{def}}{=} \exists \bar{x}. P'$ describing the current state, and an assertion $Q \overset{\text{def}}{=} \exists \bar{y}. Q'$ describing an action precondition, $\mathsf{c}\,(.,.) : \mathsf{Assn} \times \mathsf{Assn} \to \mathsf{LAssn}$, is defined as follows. We assume that P and Q are in the prenex normal form with no bound variables in P' and Q'.

$$\begin{array}{rl}\mathsf{c}\,(P, Q) \overset{\text{def}}{=} & \exists \bar{x}, \bar{y}. p \star (p' \uplus q \uplus q') \\ & \quad \text{where } (p, p') = \mathsf{ub}\,(\mathsf{f}\,(P')) \text{ and } (q, q') = \mathsf{ub}\,(\mathsf{f}\,(Q'))\end{array}$$

$$\frac{}{\mathsf{Stable}\,(p)} \qquad \frac{\mathsf{Stable}\,(P) \quad \mathsf{Stable}\,(Q)}{\mathsf{Stable}\,(P \odot Q)} \qquad \frac{\mathsf{Stable}\,(P)}{\mathsf{Stable}\,(\exists x.\, P)}$$

$$\frac{\mathsf{St}\,(P, Q) \quad \mathsf{St}\,(Q, P)}{\mathsf{Stable}\,(P * Q)}$$

$$\frac{\mathsf{Stable}\,(P)}{\mathsf{St}\,(P, R)} \qquad \frac{\mathsf{St}\,(P, Q * R) \quad \mathsf{St}\,(Q, P * R)}{\mathsf{St}\,(P * Q, R)} \qquad \frac{\mathsf{St}\,(P, R)}{\mathsf{St}\,(P, R * R')}$$

$$\frac{\mathsf{St}\,(P, R) \quad \mathsf{St}\,(Q, R)}{\mathsf{St}\,(P \odot Q, R)} \qquad \frac{\mathsf{St}\,(P, R)}{\mathsf{St}\,(\exists x.\, P, R)} \qquad \frac{\mathsf{St}\left(\boxed{p}_{I'}, R\right) \quad I' \sqsubseteq^{p} I}{\mathsf{St}\left(\boxed{p}_{I}, R\right)}$$

$$\frac{\mathsf{S}\left(p, I, R * \boxed{p}_{I}\right)}{\mathsf{St}\left(\boxed{p}_{I}, R\right)}$$

$$\frac{\mathsf{S}\,(p, I, R)}{\mathsf{S}\,(p, I, R * Q)} \qquad \frac{\mathsf{S}\,(p, I_1, R) \quad \mathsf{S}\,(p, I_2, R)}{\mathsf{S}\,(p, I_1 \cup I_2, R)} \qquad \frac{I \vdash_{\mathrm{I}} I' \quad \mathsf{S}\,(p, I', R)}{\mathsf{S}\,(p, I, R)}$$

$$\frac{\mathcal{K} * \bigotimes R \vdash_{\mathsf{SL}} \mathit{false}}{\mathsf{S}\,(p, \{\mathcal{K} : \exists \bar{y}.\, q_1 \rightsquigarrow q_2\}, R)} \qquad \frac{\mathsf{c}\,(R, q_1) \vdash_{\mathsf{SL}} \mathit{false}}{\mathsf{S}\,(p, \{\mathcal{K} : q_1 \rightsquigarrow q_2\}, R)}$$

$$\frac{(q_1 -\!\circledast \mathsf{c}\,(R, q_1)) * q_2 \vdash_{\mathsf{SL}} p * \mathit{true}}{\mathsf{S}\,(p, \{\mathcal{K} : q_1 \rightsquigarrow q_2\}, R)} \; !!!$$

Figure 2.1: Stability judgements where $P, Q, Q_1, Q_2, R \in \mathsf{FAssn}$; $p, q_1, q_2 \in \mathsf{LAssn}$ and $\dot{\in}\{\vee, *, \uplus\}$.

Guarantee We now define the guarantee relation that describes all possible updates the current thread can perform. In some sense, the guarantee relation is the dual of rely: the actions in the guarantee of one thread are included in the rely of concurrently running threads. Thus, it should come as no surprise that transitions in the guarantee can be categorised using three categories which resonate with those of the rely. The *extension* guarantee is similar to the extension rely except that new capabilities corresponding to the new shared resources are materialised in the local and shared state, and a part of the local state is moved into the shared state:

$$G^{\mathsf{e}} \stackrel{\text{def}}{=} \left\{ \begin{pmatrix} (l \circ l', g, \mathcal{I}), \\ l \circ (\mathbf{0}_{\mathbb{M}}, \kappa_1), g \circ g', \mathcal{I} \cup \mathcal{I}' \end{pmatrix} \middle| \begin{array}{l} g' = l' \circ (\mathbf{0}_{\mathbb{M}}, \kappa_2) \wedge \\ \kappa_1 \bullet_{\mathbb{K}} \kappa_2 \prec \mathsf{dom}\,(\mathcal{I}') \wedge \\ \kappa_1 \bullet_{\mathbb{K}} \kappa_2 \perp \mathsf{dom}\,(\mathcal{I}) \wedge \\ g' \circledcirc \mathcal{I}' \wedge \mathcal{I}' {\uparrow}^{(g')} (g, \mathcal{I}) \end{array} \right\}$$

The current thread may at any point extend the shared state with some of its locally held resources l'; introduce new interference to describe how the new resources may be mutated ($\mathcal{I}'$) and generate new capabilities ($\kappa_1 \bullet_{\mathbb{K}} \kappa_2 \prec \mathsf{dom}\,(\mathcal{I}')$) that facilitate the new interference, with the proviso that the new capabilities are fresh ($\kappa_1 \bullet_{\mathbb{K}} \kappa_2 \perp \mathsf{dom}\,(\mathcal{I})$). The last two conjuncts enforce closure of the new action models and can be justified as in the case of R^{e}.

The *update guarantee* G^{u} is more involved than its R^{u} counterpart, because updates in the guarantee may move resources from the local state into the shared state (similarly to extensions above) at the same time that it mutates them as prescribed by an enabled action. Intuitively, we want to enforce that resources are not created "out of thin air" in the process. This can be expressed as preserving the *orthogonal* of the combination of the local and global states, *i.e.*, the set of states compatible with that combination.

Definition 29 (Orthogonal)**.** Given any separation algebra $(\mathbb{B}, \bullet_{\mathbb{B}}, \mathbf{0}_{\mathbb{B}})$, and an element $b \in \mathbb{B}$, its *orthogonal* $(.)^{\perp}_{\mathbb{B}} : \mathbb{B} \to \mathcal{P}\,(\mathbb{B})$, is defined as the set of all elements in $\mathbb{B}$ that are compatible with it:

$$(b)^{\perp}_{\mathbb{B}} \stackrel{\text{def}}{=} \left\{ b' \mid b \sharp b' \right\}$$

The *update guarantee* G^{u} can then be defined as follows. When updating the shared state, thread are not allowed to introduce new capabilities, as this can only be achieved when extending the shared state (through G^{e}).

$$G^{\mathsf{u}} \stackrel{\text{def}}{=} \left\{ ((l, g, \mathcal{I}), (l', g', \mathcal{I})) \middle| \begin{array}{l} ((l' \circ s')_{\mathsf{K}})^{\perp}_{\mathbb{K}} = ((l \circ g)_{\mathsf{K}})^{\perp}_{\mathbb{K}} \wedge \\ \begin{pmatrix} g = g' \vee \\ \begin{pmatrix} \exists \kappa \leq l_{\mathsf{K}}.\, (g, g') \in \lceil \mathcal{I} \rceil\,(\kappa) \wedge \\ ((l \circ g)_{\mathsf{M}})^{\perp}_{\mathbb{M}} = ((l' \circ g')_{\mathsf{M}})^{\perp}_{\mathbb{M}} \end{pmatrix} \end{pmatrix} \end{array} \right\}$$

Lastly, the current thread may extend the local action model by shifting some of the existing interference. This is modelled by the G^{s} relation which is analogous to R^{s}.

$$G^{\mathsf{s}} \stackrel{\text{def}}{=} \left\{ \left((l, g, \mathcal{I}), (l, g, \mathcal{I} \cup \mathcal{I}') \right) \,\middle|\, \mathcal{I}' {\uparrow}^{(\mathbf{0}_{\mathsf{L}})} (g, \mathcal{I}) \right\}$$

Definition 30 (Guarantee)**.** The *guarantee* relation $G : \mathcal{P}(\mathsf{World} \times \mathsf{World})$ is defined as

$$G \overset{\text{def}}{=} (G^{\mathsf{u}} \cup G^{\mathsf{e}} \cup G^{\mathsf{s}})^*$$

Using the guarantee relation, we introduce the notion of *repartitioning* $P \Rrightarrow^{\{R_1\}\{R_2\}} Q$. This relation holds whenever, from any world satisfying P, if whenever parts of the composition of its local and shared states that satisfies R_1 is exchanged for one satisfying R_2, it is possible to split the resulting logical state into a local and shared part again, in such a way that the resulting transition is in G.

Definition 31 (Repartitioning)**.** We write $P \Rrightarrow^{\{R_1\}\{R_2\}} Q$ if, for every $\iota \in \mathsf{LEnv}$, and world w_1 such that $w_1, \iota \vDash P$, there exists states $m_1, m' \in \mathbb{M}$ such that $(m_1, \emptyset), \iota \vDash_{\mathsf{SL}} R_1$ and

- $m_1 \bullet_{\mathbb{M}} m' = (\downarrow(w_1))_{\mathsf{M}}$; and
- for every m_2 where $(m_2, \emptyset), \iota \vDash_{\mathsf{SL}} R_2$, there exists a world w_2 such that $w_2, \iota \vDash Q$; and
  - $m_2 \bullet_{\mathbb{M}} m' = (\downarrow(w_2))_{\mathsf{M}}$; and
  - $(w_1, w_2) \in G$

We write $P \Rrightarrow Q$ for $P \Rrightarrow^{\{\mathsf{emp}\}\{\mathsf{emp}\}} Q$, in which case the repartitioning has no "side effect" and simply shuffles resources around between the local and shared state or modifies the action models. This is the case for (Shift) and (Extend), whose proof will be given in the next section.

2.1.1 Interference Manipulations

In this section we formalise the requirements of the Extend and Shift semantic implications and show that they are valid.

Shared State Extension When extending the shared state using currently owned local resources, one specifies a new interference assertion over these newly shared resources. While in CoLoSL the new interferences may mention parts of the shared state beyond the newly added resources (in particular the existing shared state), they must not allow visible updates to those parts, so as not to invalidate other threads' views of existing resources. We thus impose a locality condition on the newly added behaviour to ensure sound extension of the shared state, similarly to the confinement constraint of local fences of Def. 7. We first motivate this constraint with an example.

Example 1. Let $P \stackrel{\text{def}}{=} x \mapsto 1 * \boxed{y \mapsto 1 \vee y \mapsto 2}_I$ denote the view of the current thread with $I \stackrel{\text{def}}{=} (\mathsf{b} : \{y \mapsto 1 \rightsquigarrow y \mapsto 2\})$. Since the current thread owns the location addressed by x, it can extend the shared state as $Q \stackrel{\text{def}}{=} [\mathsf{a}] * \boxed{(y \mapsto 1 \vee y \mapsto 2) * x \mapsto 1}_{I \cup I'}$ where $I' \stackrel{\text{def}}{=} (\mathsf{a} : x \mapsto 1 \rightsquigarrow x \mapsto 2)$. In extending the shared state, the current thread also extended the interference allowed on the shared state by adding a new action associated with the newly generated capability resource $[\mathsf{a}]$, as given in I', which updates the value of location x. Since location x was previously owned privately by the current thread and was hence not visible to other threads, this new action will not invalidate their view of the shared state, hence this extension is a valid one.

If, on the other hand, I' is replaced with $I'' \stackrel{\text{def}}{=} (\mathsf{a} : \{y \mapsto 1 \rightsquigarrow y \mapsto 3\})$, where location y can be mutated, the situation above is not allowed. Indeed, other threads may rely on the fact that the only updates allowed on location y are done through the $[\mathsf{b}]$ capability as specified in I, and would be spooked by this new possible behaviour they were not aware of (as it is not in I).

In order to ensure sound extension of the shared state, we require in EXTEND that the newly introduced interferences are confined to the locally owned resources.

Definition 32. A set of states $\mathcal{P}$ *confines* an action model $\mathcal{I}$, written $\mathcal{P} \copyright \mathcal{I}$, if

$$\exists \mathcal{F}. \mathcal{P} \subseteq \mathcal{F} \wedge \mathcal{F} \blacktriangleright \mathcal{I}$$

The $P \copyright I$ notation used in EXTEND is then a straightforward lift of this definition to assertions $P \in \mathsf{Assn}$ and interference assertions $I \in \mathsf{IAssn}$:

$$P \copyright I \stackrel{\text{def}}{\iff} \forall \iota.\ \{l \mid l, \iota \vDash_{\mathsf{SL}} P\} \copyright \langle\!\langle I \rangle\!\rangle_\iota$$

Rather than checking the confinement conditions semantically, we present a number of judgements that reduce interference confinement and local fences to logical entailments in Fig. 2.2.

Note that although local fencing judgements are given for a local assertion $f \in \mathsf{LAssn}$, since $\lfloor P \in \mathsf{LAssn}$ and $P \vdash \lfloor P$ given any assertion P (Lemma 1), in the top left confinement judgement one can simply substitute $\lfloor P$ for P. *Mutatis mutandis* for P' and Q' in the left-most judgements of the second row.

Pleasantly, since our local assertions do not contain subjective (boxed) assertions, entailments of the form $f \vdash_{\mathsf{SL}} f'$ are the familiar entailments of standard separation logic.

$$\frac{P \vdash P' \quad P' \blacktriangleright I}{P \textcircled{c}\, I} \qquad \frac{}{f \blacktriangleright \emptyset} \qquad \begin{array}{c} f \blacktriangleright I_1 \quad f \blacktriangleright I_2 \\ \hline\hline f \blacktriangleright I_1 \cup I_2 \end{array} \qquad \frac{I \vdash_{\mathrm{I}} I' \quad f \blacktriangleright I'}{f \blacktriangleright I}$$

$$\frac{\mathsf{exact}\,(r) \quad f \blacktriangleright \{\mathcal{K}{:}\, p \rightsquigarrow q\}}{f \blacktriangleright \{\mathcal{K} : p \ast r \rightsquigarrow q \ast r\}} \qquad \frac{f \uplus p \vdash_{\mathsf{SL}} \mathit{false}}{f \blacktriangleright \{\mathcal{K}{:}\, p \rightsquigarrow q\}}$$

$$\frac{(p \mathrel{-\!\!\circledast} f) \ast q \vdash_{\mathsf{SL}} f \quad f \Leftrightarrow \bigvee_{i \in I} f_i \quad (\mathsf{precise}\,(f_i) \wedge f_i \uplus p \vdash_{\mathsf{SL}} f_i \text{ for } i \in I)}{f \blacktriangleright \{\mathcal{K} : p \rightsquigarrow q\}}$$

Figure 2.2: Confinement/local fencing judgements with $P, Q \in \mathsf{Assn}$; $p, q, r, f \in \mathsf{LAssn}$.

Definition 33 (Freshness)**.** The capability assertion $\mathcal{K}$ with logical variables $\bar{x}$ is *fresh*, written $\mathit{fresh}\,(\bar{x}, \mathcal{K})$, iff for all $\iota \in \mathsf{LEnv}$ and $K \in \mathcal{P}(\mathbb{K})$ where $\diamond K$, there exists $\kappa \in \mathbb{K}$ such that

$$(\mathbf{0}_{\mathbb{M}}, \kappa), \iota \vDash_{\mathsf{SL}} \exists \bar{x}.\, \mathcal{K} \wedge \kappa \perp K$$

where

$$\kappa \perp K \overset{\text{def}}{\iff} \forall \kappa' \in K.\, \kappa \perp \kappa'$$

Lemma 4. The semantic implication (EXTEND) is valid.

Proof. It suffices to show that for all $\iota \in \mathsf{LEnv}$ and $w_1 \in \mathsf{World}$,

$$\begin{array}{ll} \text{if} & w_1, \iota \vDash P \wedge P \ast \mathcal{K}_2 \textcircled{c}\, I \wedge \mathit{fv}(P) \cap \bar{x} = \emptyset \wedge \mathit{fresh}\,(\bar{x}, \mathcal{K}_1 \ast \mathcal{K}_2) \\ \text{then} & \exists w_2.\, w_2, \iota \vDash \exists \bar{x}.\, \mathcal{K}_1 \ast \boxed{P \ast \mathcal{K}_2}_{\,I} \wedge (\downarrow(w_1))_{\mathsf{M}} = (\downarrow(w_2))_{\mathsf{M}} \wedge (w_1, w_2) \in G \end{array}$$

Pick an arbitrary $\iota \in \mathsf{LEnv}$ and $w_1 = (l, g, \mathcal{I})$ such that

$$w_1, \iota \vDash P \tag{2.8}$$
$$P \ast \mathcal{K}_2 \textcircled{c}\, I \tag{2.9}$$
$$\mathit{fresh}\,(\bar{x}, \mathcal{K}_1 \ast \mathcal{K}_2) \tag{2.10}$$

From (2.10) and the definition of *fresh* we know that

$$\exists \kappa.\, (\mathbf{0}_{\mathbb{M}}, \kappa), \iota \vDash_{\mathsf{SL}} \exists \bar{x}.\, \mathcal{K}_1 \ast \mathcal{K}_2 \;\wedge\; \kappa \perp \mathsf{dom}\,(\mathcal{I})$$

and consequently there exists $\bar{v} \in \mathsf{Val}$ and $\kappa_1, \kappa_2 \in \mathbb{K}$ such that

$$\kappa_1 \in (\!|\mathcal{K}_1[\bar{v}/\bar{x}]|\!)^K_\iota \wedge \kappa_2 \in (\!|\mathcal{K}_2[\bar{v}/\bar{x}]|\!)^K_\iota \wedge \kappa_1 \bullet_{\mathbb{K}} \kappa_2 \perp \mathsf{dom}\,(\mathcal{I}) \tag{2.11}$$

From (2.9) we have

$$P * \mathcal{K}_2[\bar{v}/\bar{x}] \copyright I[\bar{v}/\bar{x}] \tag{2.12}$$

From (2.8) and the definition of $\vDash_{\mathsf{SL}}$ we have $l, \iota \vDash_{\mathsf{SL}} P$. Similarly, from (2.11) and the definitions of $\vDash$ and $\vDash_{\mathsf{SL}}$ we have $(\mathbf{0}_{\mathbb{M}}, \kappa_2) \vDash_{\mathsf{SL}} \mathcal{K}_2$. Consequently we have:

$$l \circ (\mathbf{0}_{\mathbb{M}}, \kappa_2), \iota \vDash_{\mathsf{SL}} P * \mathcal{K}_2 \tag{2.13}$$

From the definitions of w_1 and the well-formedness of worlds we know there exists $\mathcal{F} \in \mathcal{P}(\mathsf{LState})$ such that

$$g \in \mathcal{F} \wedge \mathcal{F} \blacktriangleright \mathcal{I} \tag{2.14}$$

Pick $\mathcal{I}_0 \in \mathsf{AMod}$ such that

$$\mathsf{dom}\,(\mathcal{I}_0) = \{\kappa \mid (\kappa)^{\perp}_{\mathbb{K}} = (\kappa_1 \bullet_{\mathbb{K}} \kappa_2)^{\perp}_{\mathbb{K}}\} \wedge \forall \kappa \in \mathsf{dom}\,(\mathcal{I}_0)\,.\, \mathcal{I}_0(\kappa) = \emptyset$$

Let $\mathcal{I}'' \overset{\text{def}}{=} \langle\!| I[\bar{v}/\bar{x}] |\!\rangle_{\iota}$ and $\mathcal{I}' = \mathcal{I}'' \cup \mathcal{I}_0$; from (2.12), (2.13) and the definition of $\copyright$ we know there exists $\mathcal{F}' \in \mathcal{P}(\mathsf{LState})$ such that $l \circ (\mathbf{0}_{\mathbb{M}}, \kappa_2) \in \mathcal{F}' \wedge \mathcal{F}' \blacktriangleright \mathcal{I}''$. Consequently from the definitions of $\blacktriangleright$, $\mathcal{I}'$ and $\mathcal{I}_0$ and since $\forall \kappa \in \mathsf{dom}\,(\mathcal{I}_0)\,.\, \mathcal{I}_0(\kappa) = \emptyset$ we have

$$l \circ (\mathbf{0}_{\mathbb{M}}, \kappa_2) \in \mathcal{F}' \wedge \mathcal{F}' \blacktriangleright \mathcal{I}' \tag{2.15}$$

Let $g' = l \circ (\mathbf{0}_{\mathbb{M}}, \kappa_2)$ and $w_2 = ((\mathbf{0}_{\mathbb{M}}, \kappa_1), g \circ g', \mathcal{I} \cup \mathcal{I}')$. Given the definitions of G and G^{e}, it then suffices to show

$$(\downarrow(w_1))_{\mathsf{M}} = (\downarrow(w_2))_{\mathsf{M}} \tag{2.16}$$
$$\kappa_1 \bullet_{\mathbb{K}} \kappa_2 \prec \mathsf{dom}\,(\mathcal{I}') \tag{2.17}$$
$$\kappa_1 \bullet_{\mathbb{K}} \kappa_2 \perp \mathsf{dom}\,(\mathcal{I}) \tag{2.18}$$
$$g' \copyright \mathcal{I}' \tag{2.19}$$
$$\forall s', \mathcal{I}_0.\, \mathcal{I}{\downarrow}\,(s', g - s', \mathcal{I}_0) \implies \mathcal{I} \cup \mathcal{I}'{\downarrow}\,(s', (g - s') \circ g', \mathcal{I}_0) \tag{2.20}$$
$$w_2, \iota \vDash \exists \bar{x}.\, \mathcal{K}_1 * \boxed{P * \mathcal{K}_2}_{I} \tag{2.21}$$

RTS. (2.16) This follows immediately from the definition of $\downarrow(.)$ and the definitions of w_1 and w_2.

RTS. (2.17) This follows trivially from the definitions of $\prec$ and $\mathcal{I}'$ since $\mathcal{I}_0 \subseteq \mathcal{I}'$ and $\kappa_1 \bullet_{\mathbb{K}} \kappa_2 \in \mathsf{dom}\,(\mathcal{I}_0)$.

RTS. (2.18) This follows trivially from (2.11).

RTS. (2.19) This follows immediately from the definition of g', (2.15) and the definition of ⓒ.

RTS. (2.20) From (2.14), (2.15), the definitions of $\mathcal{I}'$, g' and Lemma 14, we can despatch this obligation.

RTS. (2.21)
From (2.11) and the definition of $\vDash_{\mathsf{SL}}$ we have $(\mathbf{0}_{\mathbb{M}}, \kappa_1), \iota \vDash_{\mathsf{SL}} \mathcal{K}_1[\bar{v}/\bar{x}]$ and consequently by the definition of $\vDash$

$$((\mathbf{0}_{\mathbb{M}}, \kappa_1), g \circ g', \mathcal{I} \cup \mathcal{I}'), \iota \vDash \mathcal{K}_1[\bar{v}/\bar{x}] \tag{2.22}$$

Similarly, from (2.11) and the definitions of $\vDash_{\mathsf{SL}}$ and $\vDash$ we have $((\mathbf{0}_{\mathbb{M}}, \kappa_2), g, \mathcal{I}), \iota \vDash \mathcal{K}_2[\bar{v}/\bar{x}]$; consequently from (2.8), the definition of g' and since $fv(P) \cap \bar{x} = \emptyset$, we can conclude

$$(g', g, \mathcal{I}), \iota \vDash (P \ast \mathcal{K}_2)[\bar{v}/\bar{x}] \tag{2.23}$$

Thus from (2.23), (2.20) - established above - and Lemma 25 we have:

$$g', \iota \vDash_{g \circ g', \mathcal{I} \cup \mathcal{I}'} (P \ast \mathcal{K}_2)[\bar{v}/\bar{x}] \tag{2.24}$$

On the other hand, from (2.14), (2.15), the definitions of $\mathcal{I}'$, g', Lemma 13 and since $\mathcal{I}' = \mathcal{I}'' \cup \mathcal{I}_0 = \langle\!|I[\bar{v}/\bar{x}]|\!\rangle_\iota \cup \mathcal{I}_0$, we have:

$$\mathcal{I} \cup \mathcal{I}' \downarrow \left(g', g, \langle\!|I[\bar{v}/\bar{x}]|\!\rangle_\iota\right) \tag{2.25}$$

Consequently from (2.24), (2.25) and the definition of $\vDash$

$$(\mathbf{0}_{\mathsf{L}}, g \circ g', \mathcal{I} \cup \mathcal{I}'), \iota \vDash (\boxed{P \ast \mathcal{K}_2}_{\,I})[\bar{v}/\bar{x}] \tag{2.26}$$

From (2.22), (2.26) and the definitions of w_2 and $\vDash$ we have:

$$w_2, \iota \vDash (\mathcal{K}_1 \ast \boxed{P \ast \mathcal{K}_2}_{\,I})[\bar{v}/\bar{x}]$$

and consequently from the definition of $\vDash$

$$w_2, \iota \vDash \exists \bar{x}.\, \mathcal{K}_1 \ast \boxed{P \ast \mathcal{K}_2}_{\,I}$$

as required. □

Action Shifting Notice that, according to our rely and guarantee conditions, the action model may only grow with time. However, that is not to say that the same holds of interference relations in shared state assertions: as seen in §**??**, they may forget about actions that are either redundant or not relevant to the current subjective view via *shifting*. As for the action model closure relation (Def. 24), this needs to be the case not only from the current subjective view but also for any evolution of it according to potential actions, both from the thread and the environment. To capture this set of possible futures, we refine our notion of local fences (Def. 10), which was defined in the context of the global shared state, to the case where we consider only a subjective state within the global shared state. For this, we need to also refine a second time our notion of action application to ignore the context of a subjective state, which as far as a subjective view is concerned could be anything.

Definition 34 (Subjective action application)**.** The *subjective application* of an action a on a logical state s, written $a(s)$ is defined provided that there exists a context r for which $a[s \circ r]$ is defined. When that is the case, we write $a(s)$ for

$$\left\{ s' \;\middle|\; \exists r.\ s \sharp r \wedge (s', -) \in a[s, r] \;\right\}$$

Note that, in contrast with $a[l]$, only *parts* of the active precondition has to intersect with the subjective view s for $a(s)$ to apply. Thus, we fabricate a context r that is compatible with the subjective view and satisfies the rest of the precondition.

Definition 35 (Fenced action model)**.** An action model $\mathcal{I} \in \mathsf{AMod}$ is *fenced* by $\mathcal{F} \in \mathcal{P}(\mathsf{LState})$, written $\mathcal{F} \rhd \mathcal{I}$, if, for all $l \in \mathcal{F}$ and all $a \in rg(\mathcal{I})$,

$$a(l) \text{ is defined} \Rightarrow a(l) \subseteq \mathcal{F}$$

In contrast with local fences, fences do not require that actions be confined inside the subjective state. We lift the notion of fences to assertions as follows, given $f \in \mathsf{Assn}$ and $I \in \mathsf{IAssn}$:

$$f \rhd I \overset{\text{def}}{\iff} \forall \iota.\, \{l \mid l, \iota \vDash_{\mathsf{SL}} F\} \rhd \langle\!\lvert I \rvert\!\rangle_\iota$$

For instance, a possible fence for the interference assertion I_{y} of Fig. **??** is denoted by the following assertion.

$$F_{\text{y}} \overset{\text{def}}{=} \bigvee_{v=0}^{10} (x \mapsto v \star y \mapsto v) \vee (x \mapsto v + 1 \star y \mapsto v)$$

Definition 36 (Action shifting)**.** Given $\mathcal{I}, \mathcal{I}' \in \mathsf{AMod}$ and $\mathcal{P} \in \mathcal{P}(\mathsf{LState})$, $\mathcal{I}'$ is a *shifting* of $\mathcal{I}$ with respect to $\mathcal{P}$, written $\mathcal{I} \sqsubseteq^{\mathcal{P}} \mathcal{I}'$, if there exists a fence $\mathcal{F}$ such that

$$\begin{array}{l}\mathcal{P} \subseteq \mathcal{F} \wedge \mathcal{F} \rhd \mathcal{I} \wedge \forall l \in \mathcal{F}. \forall \kappa. \\ (\forall a \in \mathcal{I}'(\kappa).\, \mathit{reflected}(a, l, \mathcal{I}(\kappa))) \wedge \\ \forall a \in \mathcal{I}(\kappa).\, a(l) \text{ is defined} \wedge \mathit{visible}(a, l) \Rightarrow \mathit{reflected}(a, l, \mathcal{I}'(\kappa))\end{array}$$

The first conjunct expresses the fact that the new action model $\mathcal{I}'$ cannot introduce new actions not present in $\mathcal{I}$, and the second one that $\mathcal{I}'$ has to include all the visible potential actions of $\mathcal{I}$. We lift $\sqsubseteq$ to assertions as follows:

$$I \sqsubseteq^{P} I' \stackrel{\text{def}}{=} \forall \iota.\ \langle\!\!\langle I \rangle\!\!\rangle_{\iota} \sqsubseteq^{\{s | s, \iota \vDash_{\mathsf{SL}} P\}} \langle\!\!\langle I' \rangle\!\!\rangle_{\iota}$$

We present a number of judgements that reduce action shifting and fencing conditions to logical entailments in Fig. 2.3 and 2.4. As with the local fencing judgements, the entailments in the premises of these rules are those of standard separation logic. The $\approx^{R}$ notation in the conclusion of some of the judgements denotes that shifting is valid both ways (*i.e.* $I_1 \approx^{R} I_2$ iff $I_1 \sqsubseteq^{R} I_2$ and $I_2 \sqsubseteq^{R} I_1$). We write $\mathsf{exact}(P)$ to denote that the assertion P is *exact*. That is, there exists l such that for all ι and l' where $l', \iota \vDash_{\mathsf{SL}} P$ then $l = l'$.

The premises of the form $p \cap q \vdash_{\mathsf{SL}} \mathsf{emp}$ assert that the states described by p and q *do not intersect*; that is, for all $\iota \in \mathsf{LEnv}$, and for all l_1, l_2 such that $l_1, \iota \vDash_{\mathsf{SL}} p$ and $l_2, \iota \vDash_{\mathsf{SL}} q$, then:

$$\forall l \in \mathsf{LState}.\, l \leq l_1 \wedge l \leq l_2 \implies l = \mathbf{0}_{\mathsf{L}}$$

The $p \cap q \vdash_{\mathsf{SL}} \mathsf{emp}$ entailment can be rewritten equivalently as follows:

$$p \cap q \vdash_{\mathsf{SL}} \mathsf{emp} \Leftrightarrow p \vdash_{\mathsf{SL}} \neg\left(\mathit{true} * (\neg\mathsf{emp} \wedge (\mathit{true} -\!\!\circledast\, q))\right)$$

Lemma 5. The semantic implications SHIFT is valid.

Proof. It suffices to show that for all $\iota \in \mathsf{LEnv}$ and $w_1 \in \mathsf{World}$,

$$\begin{array}{ll}\text{if} & w_1, \iota \vDash \boxed{P}_{I} \wedge \langle\!\!\langle I \rangle\!\!\rangle_{\iota} \sqsubseteq^{\{s|s,\iota\vDash_{\mathsf{SL}} P\}} \langle\!\!\langle I' \rangle\!\!\rangle_{\iota} \\ \text{then} & \exists w_2.\, w_2, \iota \vDash \boxed{P}_{I'} \wedge (\downarrow(w_1))_{\mathsf{M}} = (\downarrow(w_2))_{\mathsf{M}} \wedge (w_1, w_2) \in G\end{array}$$

Pick an arbitrary $w_1 = (l, g, \mathcal{I})$ such that

$$w_1, \iota \vDash \boxed{P}_{I} \tag{2.27}$$

$$\frac{}{\mathit{true} \rhd I} \qquad \frac{f \blacktriangleright I}{f \rhd I} \qquad \frac{f \rhd I_1 \quad f \rhd I_2}{f \rhd I_1 \cup I_2} \qquad \frac{I \vdash_{\mathrm{I}} I' \quad f \rhd I'}{f \rhd I}$$

$$\frac{\mathsf{exact}\,(r) \quad f \rhd \{\mathcal{K}{:}\, p \leadsto q\}}{f \rhd \{\mathcal{K} : p * r \leadsto q * r\}} \qquad \frac{f \rhd I' \quad I' \sqsubseteq^f I}{f \rhd I} \qquad \frac{f \cap p \vdash_{\mathsf{SL}} \mathsf{emp}}{f \rhd \{\mathcal{K}{:}\, p \leadsto q\}}$$

$$\frac{p \cap q \vdash_{\mathsf{SL}} \mathsf{emp} \quad (p \mathbin{-\!\circledast} (f \uplus p)) * q \vdash_{\mathsf{SL}} f}{f \rhd \{\mathcal{K} : p \leadsto q\}}$$

Figure 2.3: Fencing judgements.

$$\frac{P \vdash Q \quad I \sqsubseteq^Q I'}{I \sqsubseteq^P I'} \qquad \frac{f \rhd I \cup I_1 \quad I_1 \sqsubseteq^f I_2}{I \cup I_1 \sqsubseteq^f I \cup I_2} \qquad \frac{I \vdash_{\mathrm{I}} I' \quad I' \sqsubseteq^f \emptyset}{I \sqsubseteq^f \emptyset}$$

$$\frac{}{\bigcup_{i \in I} \{\mathcal{K}{:}\, \exists \bar{y}. P_i \leadsto Q\} \approx^{\mathit{true}} \left\{\mathcal{K}{:}\exists \bar{y}. \bigvee_{i \in I} P_i \leadsto Q\right\}}$$

$$\frac{}{\bigcup_{i \in I} \{\mathcal{K}{:}\, \exists \bar{y}. P \leadsto Q_i\} \approx^{\mathit{true}} \left\{\mathcal{K}{:}\exists \bar{y} P \leadsto \bigvee_{i \in I} Q_i\right\}}$$

$$\frac{}{\left\{\mathcal{K} : \exists \overline{v_i \in S_i}^{i \in I}. P \leadsto Q\right\} \approx^{\mathit{true}} \bigcup_{\overline{w_i \in S_i}^{i \in I}} \left\{\mathcal{K} : P\overline{[w_i/v_i]}^{i \in I} \leadsto Q\overline{[w_i/v_i]}^{i \in I}\right\}}$$

$$\frac{\mathsf{exact}\,(r) \quad f \cap p \vdash_{\mathsf{SL}} \mathsf{emp}}{\{\mathcal{K} : p * r \leadsto q * r\} \sqsubseteq^f \emptyset} \qquad \frac{f \uplus p \vdash_{\mathsf{SL}} \mathit{false}}{\{\mathcal{K} : p \leadsto q\} \sqsubseteq^f \emptyset}$$

$$\frac{(p \mathbin{-\!\circledast} (p \uplus f)) * q \vdash_{\mathsf{SL}} \mathit{false}}{\{\mathcal{K} : p \leadsto q\} \sqsubseteq^f \emptyset}$$

$$\frac{f \uplus p \vdash_{\mathsf{SL}} \bigvee_{i \in I} f \uplus (p * r_i) \quad \mathsf{exact}\,(r_i) \ \text{ for } i \in I}{\{\mathcal{K} : p \leadsto q\} \ \approx^f \ \bigcup_{i \in I} \{\mathcal{K} : p * r_i \leadsto q * r_i\}}$$

Figure 2.4: Action shifting judgements; we write $I \approx^f I'$ for $I \sqsubseteq^f I' \wedge I' \sqsubseteq^f I$.

$$\langle\!\lvert I\rvert\!\rangle_{\iota} \sqsubseteq^{\{s \mid s,\iota \vDash_{\mathsf{SL}} P\}} \left\langle\!\left\lvert I'\right\rvert\!\right\rangle_{\iota} \tag{2.28}$$

and let $w_2 = (l, g, \mathcal{I} \cup \langle\!\lvert I'\rvert\!\rangle_{\iota})$. Given the definition of G, we are then required to show

$$(\downarrow(w_1))_{\mathsf{M}} = (\downarrow(w_2))_{\mathsf{M}} \tag{2.29}$$
$$\forall s', \mathcal{I}_0.\, \mathcal{I}{\downarrow}\left(s', g - s', \mathcal{I}_0\right) \implies \mathcal{I} \cup \left\langle\!\left\lvert I'\right\rvert\!\right\rangle_{\iota}{\downarrow}\left(s', g - s', \mathcal{I}_0\right) \tag{2.30}$$
$$w_2, \iota \vDash \boxed{P}_{I'} \tag{2.31}$$

From (2.27) and definition of w_1 we know that there exist $s, r \in \mathsf{LState}$ such that

$$g = s \circ r \tag{2.32}$$
$$s, \iota \vDash_{g,\mathcal{I}} P \tag{2.33}$$
$$\mathcal{I}{\downarrow}\left(s, r, \langle\!\lvert I\rvert\!\rangle_{\iota}\right) \tag{2.34}$$

From the following lemma and (2.33) we know that

$$s, \iota \vDash_{\mathsf{SL}} P \tag{2.35}$$

Lemma 6. for all $P \in \mathsf{Assn}$, $\iota \in \mathsf{LEnv}$, $s, g \in \mathsf{LState}$ and $\mathcal{I} \in \mathsf{AMod}$:

$$s, \iota \vDash_{g,\mathcal{I}} P \implies s, \iota \vDash_{\mathsf{SL}} P$$

Proof. By induction on the structure of assertion P. Pick an arbitrary $\iota \in \mathsf{LEnv}$, then

Case $P \overset{\text{def}}{=} A$ Immediate.
Case $P \overset{\text{def}}{=} P_1 \implies P_2$
Pick an arbitrary $\iota \in \mathsf{LEnv}$, $s, g \in \mathsf{LState}$ and $\mathcal{I} \in \mathsf{AMod}$ such that

$$s, \iota \vDash_{g,\mathcal{I}} P_1 \implies P_2 \tag{2.36}$$
$$\forall \iota, s, g, \mathcal{I}.\, s, \iota \vDash_{g,\mathcal{I}} P_1 \implies s, \iota \vDash_{\mathsf{SL}} P_1 \tag{I.H1}$$
$$\forall \iota, s, g, \mathcal{I}.\, s, \iota \vDash_{g,\mathcal{I}} P_2 \implies s, \iota \vDash_{\mathsf{SL}} P_2 \tag{I.H2}$$

From (2.36) and the definition of $\vDash_{g,\mathcal{I}}$ we know $s, \iota \vDash_{g,\mathcal{I}} P_1$ implies $P_2 \vDash_{g,\mathcal{I}}$; consequently, from (I.H1) and (I.H2) we have: $s, \iota \vDash_{\mathsf{SL}} P_1$ implies $s, \iota \vDash_{\mathsf{SL}} P_2$. Thus, from the definition of $\vDash_{\mathsf{SL}}$ we have:

$$s, \iota \vDash_{\mathsf{SL}} P_1 \implies P_2$$

as required.

Cases $P \stackrel{\text{def}}{=} \exists x.\, P'$**;** $P \stackrel{\text{def}}{=} P_1 \star P_2$**;** $P \stackrel{\text{def}}{=} P_1 \circledast P_2$
These cases are analogous to the previous case and are omitted here.

Case $P \stackrel{\text{def}}{=} \boxed{P'}_I$
Pick an arbitrary $\iota \in \mathsf{LEnv}$, $s, g \in \mathsf{LState}$ and $\mathcal{I} \in \mathsf{AMod}$ such that

$$s, \iota \vDash_{g,\mathcal{I}} \boxed{P'}_I \tag{2.37}$$

From (2.37) and the definition of $\vDash_{g,\mathcal{I}}$ we know $s, g, \mathcal{I} \vDash \boxed{P'}_I$ and hence, from the definition of $\vDash$, we have $s = \mathbf{0}_{\mathsf{L}}$. Consequently, from the definition of $\vDash_{\mathsf{SL}}$ we have:

$$s, \iota \vDash_{\mathsf{SL}} \boxed{P'}_I$$

as required. □

Consequently, from the definition of $\sqsubseteq$, (2.28) and (2.35) we have:

$$\langle\!| I |\!\rangle_\iota \sqsubseteq^{\{s\}} \langle\!| I' |\!\rangle_\iota \tag{2.38}$$

and thus from (2.34), (2.38) and Lemma 10 we have:

$$\mathcal{I} \cup \langle\!| I' |\!\rangle_\iota \downarrow \left(s, r, \langle\!| I' |\!\rangle_\iota\right) \tag{2.39}$$

RTS. (2.29) This follows immediately from the definition of $\downarrow(.)$ and the definitions of w_1 and w_2.

RTS. (2.30) This follows immediately from the premise of the goal, (2.32), (2.34), (2.38) and Lemma 11.

RTS. (2.31) From 2.33, 2.30 (established above), and Lemma 23 we have

$$s, \iota \vDash_{g, \mathcal{I} \cup \langle\!| I' |\!\rangle_\iota} P \tag{2.40}$$

Consequently, from 2.32, 2.39, 2.40 and the definitions of $\vDash$ and w_2 we have

$$w_2, \iota \vDash \boxed{P}_{I'}$$

as required. □

2.2 Programming Language and Proof System

We define the CoLoSL proof system for deriving local Hoare triples for a simple concurrent imperative programming language. The proof system and programming language of CoLoSL are defined as an instantiation of the views framework [5].

Programming Language The CoLoSL programming language is that of the views programming language [5] instantiated with a set of *atomic* commands; It consists of skip, sequential composition, branching, loops and parallel composition. The set of CoLoSL atomic commands comprises an *atomic* construct $\langle . \rangle$ enforcing an atomic behaviour when instantiated with any *sequential* command. The sequential commands of CoLoSL are composed of a set of *elementary* commands, skip, sequential composition, branching and loops excluding atomic constructs and parallel composition. That is, atomic commands cannot be nested or contain parallel composition. CoLoSL is parametric in the choice of elementary commands allowing for suitable *instantiation* of CoLoSL depending on the programs being verified. For instance, in the token ring example of § **??** the set of elementary commands comprises variable lookup and assignment. We proceed with the formalisation of CoLoSL programming language.

Parameter 5 (Elementary commands)**.** Assume a set of elementary commands Elem, ranged over by $\mathbb{E}, \mathbb{E}_1, \cdots, \mathbb{E}_n$.

Parameter 6 (Elementary command axioms)**.** Given the separation algebra of machine states $(\mathbb{M}, \bullet_{\mathbb{M}}, \mathbf{0}_{\mathbb{M}})$, assume a set of axioms associated with elementary commands:

$$\mathrm{Ax}_{\mathsf{E}} : \mathcal{P}(\mathbb{M}) \times \mathsf{Elem} \times \mathcal{P}(\mathbb{M})$$

Definition 37 (Sequential commands)**.** Given the set of elementary commands Elem, the set of sequential commands Seqs, ranged over by $\mathbb{S}, \mathbb{S}_1, \cdots, \mathbb{S}_n$ is defined inductively as:

$$\mathbb{S} ::= \mathbb{E} \mid \mathsf{skip} \mid \mathbb{S}_1; \mathbb{S}_2 \mid \mathbb{S}_1 + \mathbb{S}_2 \mid \mathbb{S}^{\star}$$

Definition 38 (Sequential command axioms)**.** Given the axiomatisation of elementary commands Ax_{E}, the *axioms of sequential commands*:

$$\mathrm{Ax}_{\mathsf{S}} : \mathcal{P}(\mathbb{M}) \times \mathsf{Seqs} \times \mathcal{P}(\mathbb{M})$$

is defined as follows where we write $M, M', M'', \cdots$ to quantify over the elements of $\mathcal{P}(\mathbb{M})$.

$$
\begin{array}{rcl}
\mathrm{Ax}_{\mathsf{S}} & \overset{\text{def}}{=} & \mathrm{Ax}_{\mathsf{E}} \cup A_{\text{skip}} \cup A_{Seq} \cup A_{Choice} \cup A_{Rec} \\
A_{\text{skip}} & \overset{\text{def}}{=} & \{(M, \text{skip}, M) \mid M \in \mathcal{P}(\mathbb{M})\} \\
A_{Seq} & \overset{\text{def}}{=} & \left\{ (M, \mathbb{S}_1; \mathbb{S}_2, M') \;\middle|\; \begin{array}{l} (M, \mathbb{S}_1, M'') \in \mathrm{Ax}_{\mathsf{S}} \wedge \\ (M'', \mathbb{S}_2, M') \in \mathrm{Ax}_{\mathsf{S}} \end{array} \right\} \\
A_{Choice} & \overset{\text{def}}{=} & \left\{ (M, \mathbb{S}_1 + \mathbb{S}_2, M') \;\middle|\; \begin{array}{l} (M, \mathbb{S}_1, M') \in \mathrm{Ax}_{\mathsf{S}} \wedge \\ (M, \mathbb{S}_2, M') \in \mathrm{Ax}_{\mathsf{S}} \end{array} \right\} \\
A_{Rec} & \overset{\text{def}}{=} & \left\{ (M, \mathbb{S}^\star, M) \;\middle|\; (M, \mathbb{S}, M) \in \mathrm{Ax}_{\mathsf{S}} \right\}
\end{array}
$$

Definition 39 (Atomic commands)**.** Given the set of sequential commands Seqs, the set of*atomic commands* Atom, ranged over by $\mathbb{A}, \mathbb{A}_1, \cdots, \mathbb{A}_n$ is:

$$\mathbb{A} ::= \langle \mathbb{S} \rangle$$

Definition 40 (Atomic command axioms)**.** Given the axioms of sequential commands Ax_{S}, the *axioms of atomic commands*:

$$\mathrm{Ax}_{\mathsf{A}} : \mathcal{P}(\mathsf{World}) \times \mathsf{Atom} \times \mathcal{P}(\mathsf{World})$$

is defined as follows where we write $W, W', \cdots$ to quantify over the elements of $\mathcal{P}(\mathsf{World})$.

$$\mathrm{Ax}_{\mathsf{A}} \overset{\text{def}}{=} \left\{ (W, \langle \mathbb{S} \rangle, W') \;\middle|\; \begin{array}{l} (M_1, \mathbb{S}, M_2) \in \mathrm{Ax}_{\mathsf{S}} \wedge \\ W \Rrightarrow^{\{M_1\}\{M_2\}} W' \end{array} \right\}$$

Definition 41 (Programming language)**.** Given the set of atomic commands Atom, the set of CoLoSL *commands* Comm, ranged over by $\mathbb{C}, \mathbb{C}_1, \cdots \mathbb{C}_n$, is defined by the following grammar.

$$\mathbb{C}_: := \mathbb{A} \mid \mathtt{skip} \mid \mathbb{C}_1; \mathbb{C}_2 \mid \mathbb{C}_1 + \mathbb{C}_2 \mid \mathbb{C}_1 \parallel \mathbb{C}_2 \mid \mathbb{C}^\star$$

Proof Rules Our proof rules are of the form $\vdash \{P\}\ \mathbb{C}\ \{Q\}$ and carry an implicit assumption that the pre- and post-conditions of their judgements are stable. Since we build CoLoSL as an instantiation of the views framework [5], our proof rules correspond to those of [5] with the atomic commands axiomatised as per Def. 40.

Definition 42 (Proof rules)**.** The *proof rules* of CoLoSL are as described below.

$$\frac{}{\vdash \{P\} \text{ skip } \{P\}}\ \textsc{Skip} \qquad \frac{\forall \iota.\ (\llbracket P \rrbracket_\iota,\ \mathbb{A},\ \llbracket Q \rrbracket_\iota) \in \text{Ax}_{\mathsf{A}}}{\vdash \{P\}\ \mathbb{A}\ \{Q\}}\ \textsc{Atom}$$

$$\frac{\vdash \{P\}\mathbb{C}_1\{R\} \quad \vdash \{R\}\mathbb{C}_2\{Q\}}{\vdash \{P\}\ \mathbb{C}_1;\mathbb{C}_2\ \{Q\}}\ \textsc{Seq} \qquad \frac{\vdash \{P_1\}\mathbb{C}_1\{Q_1\} \quad \vdash \{P_2\}\mathbb{C}_2\{Q_2\}}{\vdash \{P_1 * P_2\}\ \mathbb{C}_1 \parallel \mathbb{C}_2\ \{Q_1 * Q_2\}}\ \textsc{Par}$$

$$\frac{\vdash \{P\}\ \mathbb{C}\ \{Q\}}{\vdash \{P * R\}\,\mathbb{C}\,\{Q * R\}}\ \textsc{Frame} \qquad \frac{P \Rrightarrow P' \quad \vdash \{P'\}\,\mathbb{C}\,\{Q'\} \quad Q' \Rrightarrow Q}{\vdash \{P\}\ \mathbb{C}\ \{Q\}}\ \textsc{Conseq}$$

$$\frac{\vdash \{P\}\mathbb{C}_1\{Q\} \quad \vdash \{P\}\mathbb{C}_2\{Q\}}{\vdash \{P\}\ \mathbb{C}_1 + \mathbb{C}_2\ \{Q\}}\ \textsc{Choice} \qquad \frac{\vdash \{P\}\,\mathbb{C}\,\{P\}}{\vdash \{P\}\ \mathbb{C}^*\ \{P\}}\ \textsc{Rec}$$

Most proof rules are standard from disjoint concurrent separation logic [14]. In the Conseq judgement, $\Rrightarrow$ denotes the repartitioning of the state as described in Def. 31.

2.3 Operational Semantics

We define the operational semantics of CoLoSL in terms of a set of *concrete* (low-level) states. Recall that the states of CoLoSL, namely worlds, are parametric in the separation algebra of machine states $(\mathbb{M}, \bullet_{\mathbb{M}}, \mathbf{0}_{\mathbb{M}})$. As such, we require that the choice of concrete states is also parametrised in CoLoSL. Since CoLoSL is an instantiation of the views framework, its operational semantics is as defined in [5] provided with the set of concrete states and the *semantics of atomic commands*. The semantics of atomic commands are defined in terms of the *interpretation of sequential and elementary commands*. Finally, as CoLoSL can be instantiated with any set of elementary commands Elem, the interpretation of elementary commands is also a parameter in CoLoSL. We proceed with the formalisation of the ingredients necessary for defining the operational semantics of atomic commands.

Parameter 7 (Concrete states)**.** Assume a set of concrete states $\mathcal{S}$ ranged over by $\sigma, \sigma_1, \cdots, \sigma_n$.

Parameter 8 (Elementary command interpretation)**.** Given the set of concrete states $\mathcal{S}$, assume an *elementary interpretation function* associating each elementary command with a non-deterministic state transformer:

$$\llbracket . \rrbracket_{\mathsf{E}}\,(.) : \mathsf{Elem} \to \mathcal{S} \to \mathcal{P}\,(\mathcal{S})$$

We lift the interpretation function to a set of concrete states such that for $S \in \mathcal{P}(\mathcal{S})$:

$$\llbracket \mathbb{E} \rrbracket_{\mathsf{E}}(S) \overset{\text{def}}{=} \bigcup_{\sigma \in S} (\llbracket \mathbb{E} \rrbracket_{\mathsf{E}}(\sigma))$$

Definition 43 (Sequential command interpretation)**.** Given the interpretation function of elementary commands $\llbracket . \rrbracket_{\mathsf{E}}(.)$, the *interpretation function for sequential commands*:

$$\llbracket . \rrbracket_{\mathsf{S}}(.) : \mathsf{Seqs} \to \mathcal{S} \to \mathcal{P}(\mathcal{S})$$

is defined inductively over the structure of sequential commands as follows:

$$\begin{array}{rcl}
\llbracket \mathbb{E} \rrbracket_{\mathsf{S}}(\sigma) & \overset{\text{def}}{=} & \llbracket \mathbb{E} \rrbracket_{\mathsf{E}}(\sigma) \\
\llbracket \mathrm{skip} \rrbracket_{\mathsf{S}}(\sigma) & \overset{\text{def}}{=} & \{\sigma\} \\
\llbracket \mathbb{S}_1 ; \mathbb{S}_2 \rrbracket_{\mathsf{S}}(\sigma) & \overset{\text{def}}{=} & \{\sigma_2 \mid S = \llbracket \mathbb{S}_1 \rrbracket_{\mathsf{S}}(\sigma) \ \wedge \ \sigma_2 \in \llbracket \mathbb{S}_2 \rrbracket_{\mathsf{S}}(S)\} \\
\llbracket \mathbb{S}_1 + \mathbb{S}_2 \rrbracket_{\mathsf{S}}(\sigma) & \overset{\text{def}}{=} & \llbracket \mathbb{S}_1 \rrbracket_{\mathsf{S}}(\sigma) \cup \llbracket \mathbb{S}_2 \rrbracket_{\mathsf{S}}(\sigma) \\
\llbracket \mathbb{S}^{\star} \rrbracket_{\mathsf{S}}(\sigma) & \overset{\text{def}}{=} & \llbracket \mathrm{skip} + \mathbb{S}; \mathbb{S}^{\star} \rrbracket_{\mathsf{S}}(\sigma)
\end{array}$$

where we lift the interpretation function to a set of concrete states such that for $S \in \mathcal{P}(\mathcal{S})$:

$$\llbracket \mathbb{S} \rrbracket_{\mathsf{S}}(S) \overset{\text{def}}{=} \bigcup_{\sigma \in S} (\llbracket \mathbb{S} \rrbracket_{\mathsf{S}}(\sigma))$$

Definition 44 (Atomic Command Interpretation)**.** Given the sequential command interpretation function $\llbracket . \rrbracket_{\mathsf{S}}(.)$, the *interpretation function for atomic commands*:

$$\llbracket . \rrbracket_{\mathsf{A}}(.) : \mathsf{Atom} \to \mathcal{S} \to \mathcal{P}(\mathcal{S})$$

is defined as :

$$\llbracket \langle \mathbb{S} \rangle \rrbracket_{\mathsf{A}}(\sigma) \overset{\text{def}}{=} \llbracket \mathbb{S} \rrbracket_{\mathsf{S}}(\sigma)$$

We lift the interpretation function to a set of concrete states such that for $S \in \mathcal{P}(\mathcal{S})$:

$$\llbracket \mathbb{S} \rrbracket_{\mathsf{A}}(S) \overset{\text{def}}{=} \bigcup_{\sigma \in S} (\llbracket \mathbb{S} \rrbracket_{\mathsf{A}}(\sigma))$$

2.4 Soundness

In order to establish the soundness of CoLoSL program logic, we relate its proof judgements to its operational semantics. To this end, we relate the

CoLoSL states, *i.e.* worlds, to the concrete states by means of a *reification function*. The reification of worlds is defined in terms of relating (reifying) machine states in $\mathbb{M}$ to concrete states in $\mathcal{S}$. As such, given that CoLoSL is parametric in the choice of underlying machine states, the *reification of machine states* is also a parameter in CoLoSL.

Since CoLoSL is an instantiation of the views framework [5], its soundness follows immediately from the soundness of views, provided that the atomic axioms are sound with respect to their operational semantics. Recall that the axioms of atomic commands are defined in terms of the axioms pertaining to elementary commands which are parametrised in CoLoSL. Thus, to ensure the soundness of atomic commands, we require that the elementary axioms are also sound with respect to their operational semantics. We proceed with

Parameter 9 (Machine state reification)**.** Given the separation algebra of machine states $(\mathbb{M}, \bullet_{\mathbb{M}}, \mathbf{0}_{\mathbb{M}})$, assume a *reification function* $\lfloor . \rfloor_{\mathbb{M}} : \mathbb{M} \to \mathcal{P}(\mathcal{S})$, relating machine states to sets of concrete states.

Parameter 10 (Elementary soundness)**.** Given the separation algebra of machine states $(\mathbb{M}, \bullet_{\mathbb{M}}, \mathbf{0}_{\mathbb{M}})$,and their reification function $\lfloor . \rfloor_{\mathbb{M}}$, assume for each elementary command $\mathbb{E} \in \mathsf{Elem}$, its axiom $(M_1, \mathbb{E}, M_2) \in \mathrm{Ax}_{\mathsf{E}}$ and any given machine state $m \in \mathbb{M}$ the following *soundness* property holds.

$$\llbracket \mathbb{E} \rrbracket_{\mathsf{E}} \left(\lfloor M_1 \bullet_{\mathbb{M}} \{m\} \rfloor_{\mathbb{M}} \right) \subseteq \lfloor M_2 \bullet_{\mathbb{M}} \{m\} \rfloor_{\mathbb{M}}$$

Definition 45 (Reification)**.** The *reification of worlds*, $\lfloor . \rfloor_W : \mathsf{World} \to \mathcal{P}(\mathcal{S})$ is defined as:

$$\lfloor (l, g, \mathcal{I}) \rfloor_W \stackrel{\text{def}}{=} \lfloor (l \circ g)_{\mathsf{M}} \rfloor_{\mathbb{M}}$$

Theorem 1 (Atomic command soundness)**.** For all $\mathbb{A} \in \mathsf{Atom}$, $(W_1, \mathbb{A}, W_2) \in \mathrm{Ax}_{\mathsf{A}}$ and $w \in \mathsf{World}$:

$$\llbracket \mathbb{A} \rrbracket_{\mathsf{A}} \left(\lfloor W_1 \bullet \{w\} \rfloor_W \right) \subseteq \lfloor W_2 \bullet_{\mathbb{M}} R\left(\{w\}\right) \rfloor_W$$

Proof. By induction over the structure of $\mathbb{A}$.

Case $\langle \mathbb{S} \rangle$
Pick an arbitrary $\mathbb{S} \in \mathsf{Seqs}$, $w \in \mathsf{World}$ and $W_1, W_2 \in \mathcal{P}(\mathsf{World})$ such that $(W_1, \langle \mathbb{S} \rangle, W_2) \in \mathrm{Ax}_{\mathsf{A}}$. From the definition of Ax_{A} we then know there exists $M_1, M_2 \in \mathcal{P}(\mathbb{M})$ such that:

$$(M_1, \mathbb{S}, M_2) \in \mathrm{Ax}_{\mathsf{S}} \land W_1 \Rrightarrow^{\{M_1\}\{M_2\}} W_2 \tag{2.41}$$

RTS.

$$\llbracket \langle \mathbb{S} \rangle \rrbracket_{\mathsf{A}} \left(\lfloor W_1 \bullet \{w\} \rfloor_W \right) \subseteq \lfloor W_2 \bullet R\left(\{w\}\right) \rfloor_W$$

Proof. Pick an arbitrary $w_1 = (l, g, \mathfrak{I}) \in W_1$; it then suffices to show that there exists $w_2 \in W_2$ and $w' \in R(w)$ such that

$$\llbracket \langle \mathbb{S} \rangle \rrbracket_{\mathsf{A}} \left(\lfloor w_1 \bullet w \rfloor_W \right) = \lfloor w_2 \bullet w' \rfloor_W \tag{2.42}$$

Given $w_1 = (l_1, g_1, \mathfrak{I}_1)$, from the definitions of $\llbracket . \rrbracket_{\mathsf{A}}(.)$ and $\lfloor . \rfloor_W$, and the properties of $\bullet$ and $\bullet_{\mathbb{M}}$ we have:

$$\begin{aligned} \llbracket \langle \mathbb{S} \rangle \rrbracket_{\mathsf{A}} \left(\lfloor w_1 \bullet w \rfloor_W \right) &= \llbracket \mathbb{S} \rrbracket_{\mathsf{S}} \left(\lfloor w_1 \bullet w \rfloor_W \right) \\ &= \llbracket \mathbb{S} \rrbracket_{\mathsf{S}} \left(\lfloor (l_1 \circ g_1)_{\mathsf{M}} \bullet_{\mathbb{M}} (w_{\mathsf{L}})_{\mathsf{M}} \rfloor_{\mathbb{M}} \right) \end{aligned} \tag{2.43}$$

On the other hand, from (2.41) and the definition of $\Rrightarrow$ we know there exists $m_1 \in M_1$ and $m' \in \mathbb{M}$ such that

$$m_1 \circ m' = (l_1 \circ g_1)_{\mathsf{M}} \wedge \tag{2.44}$$

$$\begin{aligned} &\forall m_2 \in M_2.\, \exists w_2 = (l_2, g_2, \mathfrak{I}_2) \in W_2. \\ &\quad m_2 \circ m' = (l_2 \circ g_2)_{\mathsf{M}} \wedge (w_1, w_2) \in G \end{aligned} \tag{2.45}$$

Consequently from (2.43) and (2.44) we have:

$$\llbracket \langle \mathbb{S} \rangle \rrbracket_{\mathsf{A}} \left(\lfloor w_1 \bullet w \rfloor_W \right) = \llbracket \mathbb{S} \rrbracket_{\mathsf{S}} \left(\lfloor m_1 \bullet_{\mathbb{M}} m' \bullet_{\mathbb{M}} (w_{\mathsf{L}})_{\mathsf{M}} \rfloor_{\mathbb{M}} \right) \tag{2.46}$$

From (2.41) and Lemma 16 we can rewrite (2.46) as

$$\llbracket \langle \mathbb{S} \rangle \rrbracket_{\mathsf{A}} \left(\lfloor w_1 \bullet w \rfloor_W \right) \subseteq \lfloor M_2 \bullet_{\mathbb{M}} \{ m' \bullet_{\mathbb{M}} (w_{\mathsf{L}})_{\mathsf{M}} \} \rfloor_{\mathbb{M}}$$

That is, there exits $m_2 \in M_2$ such that

$$\llbracket \langle \mathbb{S} \rangle \rrbracket_{\mathsf{A}} \left(\lfloor w_1 \bullet w \rfloor_W \right) = \lfloor m_2 \bullet_{\mathbb{M}} m' \bullet_{\mathbb{M}} (w_{\mathsf{L}})_{\mathsf{M}} \rfloor_{\mathbb{M}} \tag{2.47}$$

From (2.45) we know there exists $w_2 \in \mathsf{World}$ such that

$$w_2 = (l_2, g_2, \mathfrak{I}_2) \in W_2 \wedge m_2 \circ m' = (l_2 \circ g_2)_{\mathsf{M}} \tag{2.48}$$

$$(w_1, w_2) \in G \tag{2.49}$$

From the definition of $\lfloor . \rfloor_W$ and the properties of $\bullet_{\mathbb{M}}$ and $\bullet$ we can thus rewrite (2.47) as

$$\llbracket \langle \mathbb{S} \rangle \rrbracket_{\mathsf{A}} \left(\lfloor w_1 \bullet w \rfloor_W \right) = \lfloor (l_2 \circ w_{\mathsf{L}}, g_2, \mathfrak{I}_2) \rfloor_W \tag{2.50}$$

From (2.49) and Lemma 17 we know there exists $w' \in \mathsf{World}$ such that

$$w' = (w_{\mathsf{L}}, g_2, \mathfrak{I}_2) \wedge w' \in R(w) \tag{2.51}$$

Consequently, from (2.48), (2.50) and (2.51) we know there exists $w_2 \in W_2$ and $w' \in R(w)$ such that

$$\llbracket \langle \mathbb{S} \rangle \rrbracket_{\mathsf{A}} \left(\lfloor w_1 \bullet w \rfloor_W \right) = \lfloor w_2 \bullet w' \rfloor_W$$

as required.

□

Bibliography

[1] Richard Bornat, Cristiano Calcagno, and Hongseok Yang. Variables as resource in separation logic. *ENTCS*, 155:247–276, 2006.

[2] Cristiano Calcagno, Peter W. O'Hearn, and Hongseok Yang. Local action and abstract separation logic. In *LICS*, pages 366–378, 2007.

[3] Pedro da Rocha Pinto, Thomas Dinsdale-Young, and Philippa Gardner. TaDA: A logic for time and data abstraction. In *ECOOP*, 2014.

[4] Edsger W. Dijkstra. Self-stabilizing systems in spite of distributed control. *Commun. ACM*, 17(11):643–644, 1974.

[5] Thomas Dinsdale-Young, Lars Birkedal, Philippa Gardner, Matthew J. Parkinson, and Hongseok Yang. Views: compositional reasoning for concurrent programs. In *POPL*, pages 287–300, 2013.

[6] Thomas Dinsdale-Young, Mike Dodds, Philippa Gardner, Matthew J. Parkinson, and Viktor Vafeiadis. Concurrent abstract predicates. In Theo D'Hondt, editor, *ECOOP*, volume 6183 of *LNCS*, pages 504–528. Springer, 2010.

[7] Mike Dodds, Xinyu Feng, Matthew J. Parkinson, and Viktor Vafeiadis. Deny-guarantee reasoning. In Giuseppe Castagna, editor, *ESOP*, volume 5502 of *Lecture Notes in Computer Science*, pages 363–377. Springer, 2009.

[8] Xinyu Feng. Local rely-guarantee reasoning. In Zhong Shao and Benjamin C. Pierce, editors, *POPL*, pages 315–327. ACM, 2009.

[9] Philippa Gardner, Sergio Maffeis, and Gareth David Smith. Towards a program logic for JavaScript. In John Field and Michael Hicks, editors, *POPL*, pages 31–44. ACM, 2012.

[10] Aquinas Hobor and Jules Villard. The ramifications of sharing in data structures. In *POPL*, pages 523–536, 2013.

[11] Samin S. Ishtiaq and Peter W. O'Hearn. BI as an assertion language for mutable data structures. In *POPL*, pages 14–26, 2001.

[12] Cliff B. Jones. Specification and design of (parallel) programs. In *IFIP Congress*, pages 321–332, 1983.

[13] Ruy Ley-Wild and Aleksandar Nanevski. Subjective auxiliary state for coarse-grained concurrency. In *Proceedings of the 40th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 561–574. ACM, 2013.

[14] Peter W. O'Hearn. Resources, concurrency, and local reasoning. *Theor. Comput. Sci.*, 375(1-3):271–307, 2007.

[15] John C. Reynolds. Separation logic: A logic for shared mutable data structures. In *LICS*, pages 55–74. IEEE Computer Society, 2002.

[16] John C. Reynolds. A short course on separation logic. http://www.cs.cmu.edu/afs/cs.cmu.edu/project/fox-19/member/jcr/wwwaac2003/notes7.ps, 2003.

[17] Kasper Svendsen and Lars Birkedal. Impredicative concurrent abstract predicates. In Zhong Shao, editor, *ESOP*, volume 8410 of *Lecture Notes in Computer Science*, pages 149–168. Springer, 2014.

[18] Viktor Vafeiadis and Matthew J. Parkinson. A marriage of rely/guarantee and separation logic. In Luís Caires and Vasco Thudichum Vasconcelos, editors, *CONCUR*, volume 4703 of *LNCS*, pages 256–271. Springer, 2007.

A. Auxiliary Lemmata

Lemma 7 (FORGET-Closure)**.** For all $\mathcal{I}, \mathcal{I}' \in \mathsf{AMod}$ and $s_1, s_2, r \in \mathsf{LState}$

$$\mathcal{I}{\downarrow} (s_1 \circ s_2, r, \mathcal{I}') \implies \mathcal{I}{\downarrow} (s_1, s_2 \circ r, \mathcal{I}')$$

Proof. Pick an arbitrary $\mathcal{I}, \mathcal{I}' \in \mathsf{AMod}$ and $s_1, s_2, r \in \mathsf{LState}$ such that

$$\mathcal{I}{\downarrow} \left(s_1 \circ s_2, r, \mathcal{I}'\right) \tag{A.1}$$

From the definition of $\downarrow$, it then suffices to show

$$\mathcal{I}' \subseteq \mathcal{I} \tag{A.2}$$

$$\forall n \in \mathbb{N}.\, \mathcal{I}{\downarrow_n} \left(s_1, s_2 \circ r, \mathcal{I}'\right) \tag{A.3}$$

RTS. (A.2)
This follows trivially from (A.1) and the definition of $\downarrow$.

RTS. (A.3)
Rather than proving (A.3) directly, we first establish the following.

$$\begin{aligned} &\forall n \in \mathbb{N}.\, \forall s_1, s_2, r \in \mathsf{LState}. \\ &\qquad \mathcal{I}{\downarrow_n} \left(s_1 \circ s_2, r, \mathcal{I}'\right) \implies \mathcal{I}{\downarrow_n} \left(s_1, s_2 \circ r, \mathcal{I}'\right) \end{aligned} \tag{A.4}$$

We can then despatch (A.3) from (A.1) and (A.4); since for an arbitrary $n \in \mathbb{N}$, from (A.1) and the definition of $\downarrow$ we have $\mathcal{I}{\downarrow_n} (s_1 \circ s_2, r, \mathcal{I}')$ and consequently from (A.4) we derive $\mathcal{I}{\downarrow_n} (s_1, s_2 \circ r, \mathcal{I}')$ as required.

RTS. (A.4)
We proceed by induction on the number of steps n.

Base case $n = 0$
Pick an arbitrary $s_1, s_2, r \in \mathsf{LState}$. We are then required to show $\mathcal{I}{\downarrow_0}$

$(s_1, s_2 \circ r, \mathcal{I}')$ which follows trivially from the definition of $\downarrow_0$.

Inductive Step Pick an arbitrary $n \in \mathbb{N}$ and $s_1, s_2, r \in \mathsf{LState}$ such that

$$\mathcal{I}\downarrow_n \left(s_1 \circ s_2, r, \mathcal{I}'\right) \tag{A.5}$$

$$\forall s_1, s_2, r \in \mathsf{LState}.$$
$$\mathcal{I}\downarrow_{(n-1)} \left(s_1 \circ s_2, r, \mathcal{I}'\right) \implies \mathcal{I}\downarrow_{(n-1)} \left(s_1, s_2 \circ r, \mathcal{I}'\right) \tag{I.H.}$$

RTS.

$$\forall \kappa. \forall a \in \mathcal{I}(\kappa).\, \mathit{potential}(a, s_1 \circ s_2 \circ r) \Rightarrow$$
$$(\mathit{reflected}(a, s_1 \circ s_2 \circ r, \mathcal{I}'(\kappa)) \vee \neg \mathit{visible}(a, s_1)) \tag{A.6}$$
$$\wedge\, \forall (s', r') \in a[s_1, s_2 \circ r].\, \mathcal{I}\downarrow_{(n-1)} \left(s', r', \mathcal{I}'\right)) \tag{A.7}$$

RTS. (A.6)
Pick an arbitrary κ and $a \in \mathcal{I}(\kappa)$ such that

$$\mathit{potential}(a, s_1 \circ s_2 \circ r)$$

Then from (A.5) we have:

$$\mathit{reflected}(a, s_1 \circ s_2 \circ r, \mathcal{I}'(\kappa)) \vee \neg \mathit{visible}(a, s_1 \circ s_2)$$

In the case of the first disjunct the desired result holds trivially. On the other hand in the case of the second disjunct from the definition of *visible* we have $\neg \mathit{visible}(a, s_1)$ as required.

RTS. (A.7)
Pick an arbitrary κ and $a = (p, q, c) \in \mathcal{I}(\kappa)$ such that

$$\mathit{potential}(a, s_1 \circ s_2 \circ r)$$

From (A.5) we then have:

$$\forall (s', r') \in a[s_1 \circ s_2, r].\, \mathcal{I}\downarrow_{(n-1)} \left(s', r', \mathcal{I}'\right) \tag{A.8}$$

Pick an arbitrary (s', r') such that

$$(s', r') \in a[s_1, s_2 \circ r] \tag{A.9}$$

Then from the definition of $a[s_1, s_2 \circ r]$ and by the cross-split property we know there exists $ps_1, ps_2, p_r, s_1', s_2', r''' \in \mathsf{LState}$ such that :

$$\begin{array}{l} p = ps_1 \circ ps_2 \circ p_r \wedge s_1 = ps_1 \circ s_1' \wedge s_2 = ps_2 \circ s_2' \wedge r = p_r \circ r'' \wedge \\ \begin{pmatrix} (ps_1 > \mathbf{0}_\mathsf{L} \wedge s' = q \circ s_1' \wedge r' = s_2' \circ r'') \\ \vee\, (ps_1 = \mathbf{0}_\mathsf{L} s' = s_1 \wedge r' = q \circ s_2' \circ r'') \end{pmatrix} \end{array} \tag{A.10}$$

and consequently from the definition of $a[s_1 \circ s_2, r]$

$$\begin{array}{l}
p = ps_1 \circ ps_2 \circ p_r \wedge s_1 = ps_1 \circ s_1' \wedge s_2 = ps_2 \circ s_2' \wedge r = p_r \circ r'' \wedge \\
\left(\begin{array}{l}
(s' = q \circ s_1' \wedge r' = s_2' \circ r'' \wedge (q \circ s_1' \circ s_2', r'') \in a[s_1 \circ s_2, r]) \\
\vee \left(\begin{array}{l}
ps_1 = \mathbf{0_L} \wedge s' = s_1 \wedge r' = q \circ s_2' \circ r'' \wedge \\
\left(\begin{array}{l}
(ps_2 = \mathbf{0_L} \wedge (s_1' \circ s_2', q \circ r'') \in a[s_1 \circ s_2, r]) \\
\vee (ps_2 > \mathbf{0_L} \wedge (s_1' \circ q \circ s_2', r'') \in a[s_1 \circ s_2, r])
\end{array}\right)
\end{array}\right)
\end{array}\right)
\end{array}$$

That is,

$$\exists s''.\, (s' \circ s'', r' - s'') \in a[s_1 \circ s_2, r] \tag{A.11}$$

From (A.8) and (A.11) we then have

$$\exists s''.\, \mathcal{I}{\downarrow_{(n-1)}} \left(s' \circ s'', r' - s'', \mathcal{I}'\right)$$

Finally from (I.H.) we have

$$\exists s''.\, \mathcal{I}{\downarrow_{(n-1)}} \left((s', (r' - s'') \circ s'', \mathcal{I}'\right)$$

That is,

$$\mathcal{I}{\downarrow_{(n-1)}} \left(s', r', \mathcal{I}'\right)$$

as required. $\square$

Lemma 8 (MERGE-Closure)**.** For all $\mathcal{I}, \mathcal{I}_1, \mathcal{I}_2 \in \mathsf{AMod}$ and $s_p, s_c, s_q, r \in \mathsf{LState}$,

$$\mathcal{I}\!\downarrow (s_p \circ s_c, s_q \circ r, \mathcal{I}_1) \wedge \mathcal{I}\!\downarrow (s_q \circ s_c, s_p \circ r, \mathcal{I}_2) \implies \mathcal{I}\!\downarrow (s_p \circ s_c \circ s_q, r, \mathcal{I}_1 \cup \mathcal{I}_2)$$

Proof. Pick an arbitrary $\mathcal{I}, \mathcal{I}_1, \mathcal{I}_2 \in \mathsf{AMod}$ and $s_p, s_c, s_q, r \in \mathsf{LState}$ such that

$$\mathcal{I}\!\downarrow (s_p \circ s_c, s_q \circ r, \mathcal{I}_1) \wedge \mathcal{I}\!\downarrow (s_q \circ s_c, s_p \circ r, \mathcal{I}_2) \tag{A.12}$$

From the definition of $\downarrow$, it then suffices to show

$$\mathcal{I}_1 \cup \mathcal{I}_2 \subseteq \mathcal{I} \tag{A.13}$$

$$\forall n \in \mathbb{N}.\, \mathcal{I}\!\downarrow_n (s_p \circ s_c \circ s_q, r, \mathcal{I}_1 \cup \mathcal{I}_2) \tag{A.14}$$

RTS. (A.13)
Since from (A.12) and the definition of $\downarrow$ we have $\mathcal{I}_1 \subseteq \mathcal{I} \wedge \mathcal{I}_2 \subseteq \mathcal{I}$, we can thus conclude $\mathcal{I}_1 \cup \mathcal{I}_2 \subseteq \mathcal{I}$ as required.

RTS. (A.14)
Rather than proving (A.14) directly, we first establish the following.

$$\begin{aligned} &\forall n \in \mathbb{N}.\, \forall s_p, s_c, s_q, r \in \mathsf{LState}. \\ &\quad \mathcal{I}\!\downarrow_n (s_p \circ s_c, s_q \circ r, \mathcal{I}_1) \wedge \mathcal{I}\!\downarrow_n (s_c \circ s_q, s_p \circ r, \mathcal{I}_2) \\ &\qquad \implies \mathcal{I}\!\downarrow_n (s_p \circ s_c \circ s_q, r, \mathcal{I}_1 \cup \mathcal{I}_2) \end{aligned} \tag{A.15}$$

We can then despatch (A.14) from (A.12) and (A.15); since for an arbitrary $n \in \mathbb{N}$, from (A.12) and the definition of $\downarrow$ we have $\mathcal{I}\!\downarrow_n (s_p \circ s_c, s_q \circ r, \mathcal{I}_1) \wedge \mathcal{I}\!\downarrow_n (s_c \circ s_q, s_p \circ r, \mathcal{I}_2)$ and consequently from (A.15) we derive $\mathcal{I}\!\downarrow_n (s_p \circ s_c \circ s_q, r, \mathcal{I}_1 \cup \mathcal{I}_2)$ as required.

RTS. (A.15)
We proceed by induction on the number of steps n.

Base case $n = 0$
Pick an arbitrary $s_1, s_2, r \in \mathsf{LState}$. We are then required to show $\mathcal{I}\downarrow_0 (s_1, s_2 \circ r, \mathcal{I}')$ which follows trivially from the definition of $\downarrow_0$.

Inductive Step Pick an arbitrary $s_p, s_q, s_c, r \in \mathsf{LState}$ and $n \in \mathbb{N}$, such that

$$\mathcal{I}\!\downarrow_n (s_p \circ s_c, s_q \circ r, \mathcal{I}_1) \tag{A.16}$$

$$\mathcal{I}\!\downarrow_n (s_q \circ s_c, s_p \circ r, \mathcal{I}_2) \tag{A.17}$$

$$\begin{array}{l}\forall s_p, s_q, s_c, r \in \mathsf{LState}. \\ \quad \mathcal{I}{\downarrow_{(n-1)}} (s_p \circ s_c, s_q \circ r, \mathcal{I}_1) \;\wedge\; \mathcal{I}{\downarrow_{(n-1)}} (s_q \circ s_c, s_p \circ r, \mathcal{I}_2) \\ \qquad\qquad \Longrightarrow \mathcal{I}{\downarrow_{(n-1)}} (s_p \circ s_c \circ s_q, r, \mathcal{I}_1 \cup \mathcal{I}_2)\end{array} \tag{I.H.}$$

RTS.

$$\begin{array}{l}\forall \kappa.\, \forall a \in \mathcal{I}(\kappa).\, \mathit{potential}(a, s_p \circ s_c \circ s_q \circ r) \Rightarrow \\ \quad (\mathit{reflected}(a, s_p \circ s_c \circ s_q \circ r, (\mathcal{I}_1 \cup \mathcal{I}_2)(\kappa)) \vee \neg \mathit{visible}(a, s_p \circ s_c \circ s_q)\;) \wedge \\ \quad \forall (s', r') \in a[s_p \circ s_c \circ s_q, r].\, \mathcal{I}{\downarrow_{(n-1)}} (s', r', \mathcal{I}_1 \cup \mathcal{I}_2))\end{array}$$

Pick an arbitrary κ and $a = (p, q, c) \in \mathcal{I}(\kappa)$ such that

$$\mathit{potential}(a, s_p \circ s_c \circ s_q \circ r) \tag{A.18}$$

From (A.16) and (A.18) we have:

$$\begin{array}{l}(\mathit{reflected}(a, s_p \circ s_c \circ s_q \circ r, \mathcal{I}_1(\kappa)) \vee \neg \mathit{visible}(a, s_p \circ s_c)) \wedge \\ \forall (s', r') \in a[s_p \circ s_c, s_q \circ r].\, \mathcal{I}{\downarrow_{(n-1)}} (s', r', \mathcal{I}_1)\end{array}$$

and consequently from the definition of $\mathcal{I}_1 \cup \mathcal{I}_2$ we have:

$$\begin{array}{l}(\mathit{reflected}(a, s_p \circ s_c \circ s_q \circ r, (\mathcal{I}_1 \cup \mathcal{I}_2)(\kappa)) \vee \neg \mathit{visible}(a, s_p \circ s_c)) \wedge \\ \forall (s', r') \in a[s_p \circ s_c, s_q \circ r].\, \mathcal{I}{\downarrow_{(n-1)}} (s', r', \mathcal{I}_1)\end{array} \tag{A.19}$$

Similarly, from (A.17) and (A.18) we have:

$$\begin{array}{l}(\mathit{reflected}(a, s_p \circ s_c \circ s_q \circ r, (\mathcal{I}_1 \cup \mathcal{I}_2)(\kappa)) \vee \neg \mathit{visible}(a, s_c \circ s_q)) \wedge \\ \forall (s', r') \in a[s_c \circ s_q, s_p \circ r].\, \mathcal{I}{\downarrow_{(n-1)}} (s', r', \mathcal{I}_2)\end{array} \tag{A.20}$$

From (A.19), (A.20) and the definition of *visible* we have:

$$\begin{array}{l}(\mathit{reflected}(a, s_p \circ s_c \circ s_q \circ r, (\mathcal{I}_1 \cup \mathcal{I}_2)(\kappa)) \vee \neg \mathit{visible}(a, s_p \circ s_c \circ s_q)) \wedge \\ \forall (s', r') \in a[s_p \circ s_c, s_q \circ r].\, \mathcal{I}{\downarrow_{(n-1)}} (s', r', \mathcal{I}_1) \wedge \\ \forall (s', r') \in a[s_c \circ s_q, s_p \circ r].\, \mathcal{I}{\downarrow_{(n-1)}} (s', r', \mathcal{I}_2)\end{array} \tag{A.21}$$

Pick an arbitrary $s', r' \in \mathsf{LState}$ such that

$$(s', r') \in a[s_p \circ s_c \circ s_q, r] \tag{A.22}$$

Then from the definition of $a[s_p \circ s_c \circ s_q]$ and by the cross-split property we know there exists $p_p, p_c, p_q, s'_p, s'_c, s'_q \in \mathsf{LState}$ such that :

$$\begin{array}{l}s' = s_p \circ s_c \circ s_q \vee \\ \left(\begin{array}{l}(p_p > \mathbf{0}_\mathsf{L} \vee p_c > \mathbf{0}_\mathsf{L} \vee p_q > \mathbf{0}_\mathsf{L}) \wedge s' = q \circ s'_p \circ s'_c \circ s'_q \\ \wedge p = p_p \circ p_c \circ p_q \circ p_r \\ \wedge s_p = p_p \circ s'_p \wedge s_c = p_c \circ s'_c \wedge s_q = p_q \circ s'_q \wedge r = p_r \circ r'\end{array} \right)\end{array} \tag{A.23}$$

and consequently from the definitions of $a[s_p \circ s_c, s_q \circ r]$ and $a[s_c \circ s_q, s_p \circ r]$ we have:

$$
\begin{array}{l}
\left(\begin{array}{l}
s' = s_p \circ s_c \circ s_q \wedge \\
(s_p \circ s_c, s_q \circ r') \in a[s_p \circ s_c, s_q \circ r] \wedge (s_c \circ s_q, s_p \circ r') \in a[s_c \circ s_q, s_p \circ r]
\end{array}\right) \\
\vee \left(\begin{array}{l}
s' = q \circ s'_p \circ s'_c \circ s'_q \wedge \\
\left(\begin{array}{l}
\left(\begin{array}{l}
((p_c > \mathbf{0}_{\mathsf{L}} \vee (p_c = \mathbf{0}_{\mathsf{L}} \wedge p_p > \mathbf{0}_{\mathsf{L}} \wedge p_q > \mathbf{0}_{\mathsf{L}})))\wedge \\
(q \circ s'_p \circ s'_c, s'_q \circ r') \in a[s_p \circ s_c, s_q \circ r] \wedge \\
(q \circ s'_c \circ s'_q, s'_p \circ r') \in a[s_c \circ s_q, s_p \circ r]
\end{array}\right) \\
\vee \left(\begin{array}{l}
p_p > \mathbf{0}_{\mathsf{L}} \wedge p_c = p_q = \mathbf{0}_{\mathsf{L}} \wedge \\
(q \circ s'_p \circ s'_c, s'_q \circ r') \in a[s_p \circ s_c, s_q \circ r] \wedge \\
(s'_c \circ s'_q, q \circ s'_p \circ r') \in a[s_c \circ s_q, s_p \circ r]
\end{array}\right) \\
\vee \left(\begin{array}{l}
p_q > \mathbf{0}_{\mathsf{L}} \wedge p_c = p_p = \mathbf{0}_{\mathsf{L}} \wedge \\
(s'_p \circ s'_c, q \circ s'_q \circ r') \in a[s_p \circ s_c, s_q \circ r] \wedge \\
(q \circ s'_c \circ s'_q, s'_p \circ r') \in a[s_c \circ s_q, s_p \circ r]
\end{array}\right)
\end{array}\right)
\end{array}\right)
\end{array}
\tag{A.24}
$$

From (A.24), (A.16), (A.17) and (A.18) we have:

$$
\begin{array}{l}
\left(\begin{array}{l}
s' = s_p \circ s_c \circ s_q \wedge \\
\mathcal{J}{\downarrow_{(n-1)}} (s_p \circ s_c, s_q \circ r', \mathcal{J}_1) \wedge \mathcal{J}{\downarrow_{(n-1)}} (s_c \circ s_q, s_p \circ r', \mathcal{J}_2)
\end{array}\right) \\
\vee \left(\begin{array}{l}
s' = q \circ s'_p \circ s'_c \circ s'_q \wedge \\
\left(\begin{array}{l}
\left(\begin{array}{l}
((p_c > \mathbf{0}_{\mathsf{L}} \vee (p_c = \mathbf{0}_{\mathsf{L}} \wedge p_p > \mathbf{0}_{\mathsf{L}} \wedge p_q > \mathbf{0}_{\mathsf{L}})))\wedge \\
\mathcal{J}{\downarrow_{(n-1)}} \left(q \circ s'_p \circ s'_c, s'_q \circ r', \mathcal{J}_1\right) \wedge \\
\mathcal{J}{\downarrow_{(n-1)}} \left(q \circ s'_c \circ s'_q, s'_p \circ r', \mathcal{J}_2\right)
\end{array}\right) \\
\vee \left(\begin{array}{l}
p_p > \mathbf{0}_{\mathsf{L}} \wedge p_c = p_q = \mathbf{0}_{\mathsf{L}} \wedge \\
\mathcal{J}{\downarrow_{(n-1)}} \left(q \circ s'_p \circ s'_c, s'_q \circ r', \mathcal{J}_1\right) \wedge \\
\mathcal{J}{\downarrow_{(n-1)}} \left(s'_c \circ s'_q, q \circ s'_p \circ r', \mathcal{J}_2\right)
\end{array}\right) \\
\vee \left(\begin{array}{l}
p_q > \mathbf{0}_{\mathsf{L}} \wedge p_c = p_p = \mathbf{0}_{\mathsf{L}} \wedge \\
\mathcal{J}{\downarrow_{(n-1)}} \left(s'_p \circ s'_c, q \circ s'_q \circ r', \mathcal{J}_1\right) \wedge \\
\mathcal{J}{\downarrow_{(n-1)}} \left(q \circ s'_c \circ s'_q, s'_p \circ r', \mathcal{J}_2\right)
\end{array}\right)
\end{array}\right)
\end{array}\right)
\end{array}
\tag{A.25}
$$

and thus from (A.25) and (I.H.)

$$
\mathcal{J}{\downarrow_{(n-1)}} \left(s', r', \mathcal{J}_1 \cup \mathcal{J}_2\right) \tag{A.26}
$$

Finally, from (A.21), (A.22) and (A.26) we have:

$$
\begin{array}{l}
(\mathit{reflected}(a, s_p \circ s_c \circ s_q \circ r, (\mathcal{J}_1 \cup \mathcal{J}_2)(\kappa)) \vee \neg \mathit{visible}(a, s_p \circ s_c \circ s_q)) \wedge \\
\forall (s', r') \in a[s_p \circ s_c \circ s_q, r].\, \mathcal{J}{\downarrow_{(n-1)}} (s', r', \mathcal{J}_1 \cup \mathcal{J}_2)
\end{array}
$$

as required.

$\square$

Lemma 9 (Shift-Fence)**.** For all $\mathcal{I}_1, \mathcal{I}_2 \in \mathsf{AMod}$, $s, s', r \in \mathsf{LState}$ and $a \in rg(\mathcal{I}_1)$:

$$\mathcal{I}_1 \sqsubseteq^{\{s\}} \mathcal{I}_2 \land \big(s' \in a(s) \lor (s', -) \in a[s, r]\big) \implies \mathcal{I}_1 \sqsubseteq^{\{s'\}} \mathcal{I}_2$$

Proof. Pick an arbitrary $\mathcal{I}_1, \mathcal{I}_2 \in \mathsf{AMod}$, $s, s', r \in \mathsf{LState}$ and $a \in rg(\mathcal{I}_1)$ such that:

$$\mathcal{I}_1 \sqsubseteq^{\{s\}} \mathcal{I}_2 \tag{A.27}$$

RTS. $\mathcal{I}_1 \sqsubseteq^{\{s'\}} \mathcal{I}_2$.
There are two cases to consider:
Case 1. $s' \in a(s)$
From the definition of $\sqsubseteq^{\{s\}}$ and (A.27) we know there exists a fence $\mathcal{F}$ such that:

$$s \in \mathcal{F} \tag{A.28}$$

$$\mathcal{F} \rhd \mathcal{I}_1 \tag{A.29}$$

$$\begin{aligned}&\forall l \in \mathcal{F}. \forall \kappa. \forall a \in \mathcal{I}_2(\kappa).\, \mathit{reflected}(a, l, \mathcal{I}_1(\kappa)) \land \\ &\qquad \forall a \in \mathcal{I}_1(\kappa).\, a(l) \text{ is defined } \land \mathit{visible}(a, l) \implies \mathit{reflected}(a, l, \mathcal{I}_2(\kappa))\end{aligned} \tag{A.30}$$

By definition of $\rhd$ and from (A.28)-(A.29) and assumption of case 1. we have:

$$s' \in \mathcal{F} \tag{A.31}$$

Finally by definition of $\sqsubseteq^{\{s'\}}$ and (A.29)-(A.31) we have

$$\mathcal{I}_1 \sqsubseteq^{\{s'\}} \mathcal{I}_2$$

as required.

Case 2. $(s', -) \in a[s, r]$
From the definitions of $a[s, r]$ and $a(s)$ and from the assumption of the case we know $s' \in a(s)$. The rest of the proof is identical to that of case 1. $\square$

Lemma 10 (Shift-Closure-1)**.** For all $\mathcal{I}_1, \mathcal{I}_2, \mathcal{I} \in \mathsf{AMod}$ and $s, r \in \mathsf{LState}$,

$$\mathcal{I}{\downarrow} (s, r, \mathcal{I}_1) \land \mathcal{I}_1 \sqsubseteq^{\{s\}} \mathcal{I}_2 \implies \mathcal{I} \cup \mathcal{I}_2{\downarrow} (s, r, \mathcal{I}_2)$$

Proof. Pick an arbitrary $\mathcal{I}_1, \mathcal{I}_2, \mathcal{I} \in \mathsf{AMod}$ and $s, r \in \mathsf{LState}$ such that

$$\mathcal{I}{\downarrow}\,(s, r, \mathcal{I}_1) \tag{A.32}$$

$$\mathcal{I}_1 \sqsubseteq^{\{s\}} \mathcal{I}_2 \tag{A.33}$$

From the definition of $\downarrow$, it then suffices to show

$$\mathcal{I}_2 \subseteq \mathcal{I} \cup \mathcal{I}_2 \tag{A.34}$$

$$\forall n \in \mathbb{N}.\, \mathcal{I} \cup \mathcal{I}_2{\downarrow_n}\,(s, r, \mathcal{I}_2) \tag{A.35}$$

RTS. (A.34)
This holds trivially from the definition of $\mathcal{I} \cup \mathcal{I}_2$.

RTS. (A.35)
Rather than proving (A.35) directly, we first establish the following.

$$\begin{aligned}&\forall n \in \mathbb{N}.\, \forall s, r \in \mathsf{LState}.\\ &\quad \mathcal{I}{\downarrow_n}\,(s, r, \mathcal{I}_1) \land \mathcal{I}_1 \sqsubseteq^{\{s\}} \mathcal{I}_2 \implies \mathcal{I} \cup \mathcal{I}_2{\downarrow_n}\,(s, r, \mathcal{I}_2)\end{aligned} \tag{A.36}$$

We can then despatch (A.35) from (A.32), (A.33) and (A.36); since for an arbitrary $n \in \mathbb{N}$, from (A.32) and the definition of $\downarrow$ we have $\mathcal{I}{\downarrow_n}\,(s, r, \mathcal{I}_1)$ and consequently from (A.33) and (A.36) we derive $\mathcal{I} \cup \mathcal{I}_2{\downarrow_n}\,(s, r, \mathcal{I}_2)$ as required.

RTS. (A.36)
We proceed by induction on the number of steps n.

Base case $n = 0$
Pick an arbitrary $s, r \in \mathsf{LState}$. We are then required to show $\mathcal{I} \cup \mathcal{I}_2{\downarrow_0}\,(s, r, \mathcal{I}_2)$ which follows trivially from the definition of $\downarrow_0$.

Inductive Case
Pick an arbitrary $s, r \in \mathsf{LState}$ such that:

$$\mathcal{I}{\downarrow_n}\,(s, r, \mathcal{I}_1) \tag{A.37}$$

$$\mathcal{I}_1 \sqsubseteq^{\{s\}} \mathcal{I}_2 \tag{A.38}$$

$$\begin{aligned}&\forall s, r \in \mathsf{LState}.\\ &\quad \mathcal{I}{\downarrow_{(n-1)}}\,(s, r, \mathcal{I}_1) \land \mathcal{I}_1 \sqsubseteq^{\{s\}} \mathcal{I}_2 \implies \mathcal{I} \cup \mathcal{I}_2{\downarrow_{(n-1)}}\,(s, r, \mathcal{I}_2)\end{aligned} \tag{I.H}$$

RTS.

$$\forall \kappa.\, \forall a \in (\mathcal{I} \cup \mathcal{I}_2)\,(\kappa).\, \mathit{potential}(a, s \circ r) \Rightarrow$$

$$\begin{array}{l}(\mathit{reflected}(a, s \circ r, \mathcal{I}_2(\kappa)) \vee \neg \mathit{visible}(a, s)\ ) \wedge \\ \forall (s', r') \in a[s, r].\, \mathcal{I} \cup \mathcal{I}_2 \downarrow_{(n-1)} (s', r', \mathcal{I}_2)\end{array} \tag{A.39}$$

Pick an arbitrary κ and $a = (p, q, c) \in (\mathcal{I} \cup \mathcal{I}_2)\,(\kappa)$ such that:

$$\mathit{potential}(a, s \circ r) \tag{A.40}$$

Since either $a \in \mathcal{I}(\kappa)$ or $a \in \mathcal{I}_2(\kappa)$, there are two cases to consider:

Case 1. $a \in \mathcal{I}(\kappa)$.
From (A.37) and (A.40) we have:

$$\begin{array}{l}(\mathit{reflected}(a, s \circ r, \mathcal{I}_1(\kappa)) \vee \neg \mathit{visible}(a, s)\ ) \wedge \\ \forall (s', r') \in a[s, r].\, \mathcal{I} \downarrow_{(n-1)} (s', r', \mathcal{I}_1)\end{array} \tag{A.41}$$

Pick an arbitrary (s', r') such that

$$(s', r') \in a[s, r] \tag{A.42}$$

Then from (A.41) we have:

$$\mathcal{I} \downarrow_{(n-1)} \left(s', r', \mathcal{I}_1\right) \tag{A.43}$$

On the other hand, from (A.42) and Lemma 9 we have:

$$\mathcal{I}_1 \sqsubseteq^{\{s'\}} \mathcal{I}_2 \tag{A.44}$$

Consequently, from (A.43), (A.44) and (I.H) we have:

$$\mathcal{I} \cup \mathcal{I}_2 \downarrow_{(n-1)} \left(s', r', \mathcal{I}_2\right)$$

and thus from (A.42) we have

$$\forall (s', r') \in a[s, r].\, \mathcal{I} \cup \mathcal{I}_2 \downarrow_{(n-1)} \left(s', r', \mathcal{I}_2\right) \tag{A.45}$$

Since either $\mathit{visible}(a, s)$ or $\neg\mathit{visible}(a, s)$, there are two cases to consider:

Case 1.1. $\neg\mathit{visible}(a, s)$
From the assumption of case 1.1. and (A.45) we then have

$$\begin{array}{l}\neg \mathit{visible}(a, s) \wedge \\ \forall (s', r') \in a[s, r].\, \mathcal{I} \cup \mathcal{I}_2 \downarrow_{(n-1)} \left(s', r', \mathcal{I}_2\right)\end{array}$$

as required.

Case 1.2. $\mathit{visible}(a, s)$
From (A.41) and the assumption of case 1.2. we then have

$$\mathit{reflected}(a, s \circ r, \mathcal{I}_1(\kappa)) \tag{A.46}$$

Pick an arbitrary $l \in \mathsf{LState}$ such that:

$$p \circ c \leq s \circ r \circ l \tag{A.47}$$

Then from (A.46) and the definition of *reflected* we have:

$$\exists a', c'.\, a' = (p, q, c') \land a' \in \mathcal{I}_1(\kappa) \land p \circ c' \leq s \circ r \circ l \tag{A.48}$$

From (A.40) and by definition of *potential* we know $a[s \circ r]$ is defined; from (A.48), and the definition of $a'[s \circ r]$ we know that $a'[s \circ r]$ is also defined. Consequently, from the definition of $a'(s)$, we know $a'(s)$ is also defined. Thus, from the definition of *visible*, (A.48) and the assumption of case 1.2 we have

$$\mathit{visible}(a', s) \tag{A.49}$$

Thus from (A.38), (A.48), (A.49) and from the definition of $\sqsubseteq^{\{s\}}$ we have

$$\exists a'', c''.\, a'' = (p, q, c'') \land a'' \in \mathcal{I}_2(\kappa) \land p \circ c'' \leq s \circ r \circ l \tag{A.50}$$

Finally, from (A.47), (A.48), (A.50) and by definition of *reflected* we have:

$$\mathit{reflected}(a, s \circ r, \mathcal{I}_2(\kappa)) \tag{A.51}$$

From (A.45) and (A.51) we have

$$\begin{array}{l} \mathit{reflected}(a, s \circ r, \mathcal{I}_2(\kappa)) \land \\ \forall (s', r') \in a[s, r].\, \mathcal{I} \cup \mathcal{I}_2 \downarrow_{(n-1)} \left(s', r', \mathcal{I}_2\right) \end{array}$$

as required.

Case 2. $a \in \mathcal{I}_2(\kappa)$.
From the assumption of the case we trivially have

$$\mathit{reflected}(a, s \circ r, \mathcal{I}_2(\kappa)) \tag{A.52}$$

Pick an arbitrary $s', r' \in \mathsf{LState}$ such that:

$$(s', r') \in a[s, r] \tag{A.53}$$

From (A.40), the definition of *potential* and by Lemma 26 we know there exists l such that

$$\begin{aligned}
&p \circ c < s \circ r \circ l \land \\
&\exists l'.\, p \circ l' = s \circ r \land q \sharp l'
\end{aligned}$$

and thus from (A.38) we know there exists c', a' such that:

$$\begin{aligned}
&a' = (p, q, c') \in \mathcal{I}_1(\kappa) \land \\
&p \circ c' < s \circ r \circ l \land \\
&\exists l'.\, p \circ l' = s \circ r \land q \sharp l'
\end{aligned}$$

and consequently from the definition of *potential* and Lemma 26 we have:

$$\mathit{potential}(a', s \circ r) \tag{A.54}$$

Since $a = (p, q, c)$ and $a' = (p, q, c')$ and from the definition of $a[s, r]$ we know $a[s, r] = a'[s, r]$. Thus, from (A.37), (A.40), (A.53) and (A.54) we have:

$$\mathcal{I}{\downarrow_{(n-1)}} \left(s', r', \mathcal{I}_1\right) \tag{A.55}$$

From (A.38), (A.53), Lemma 9 and since $a[s, r] = a'[s, r]$, we have

$$\mathcal{I}_1 \sqsubseteq^{\{s'\}} \mathcal{I}_2 \tag{A.56}$$

Finally, from (A.55), (A.56) and (I.H) we have:

$$\mathcal{I} \cup \mathcal{I}_2{\downarrow_{(n-1)}} \left(s', r', \mathcal{I}_2\right)$$

Thus from (A.53) we have

$$\forall (s', r') \in a[s, r].\, \mathcal{I} \cup \mathcal{I}_2{\downarrow_{(n-1)}} \left(s', r', \mathcal{I}_2\right) \tag{A.57}$$

and consequently, from (A.52) and (A.57) we have

$$\begin{aligned}
&\mathit{reflected}(a, s \circ r, \mathcal{I}_2(\kappa)) \land \\
&\forall (s', r') \in a[s, r].\, \mathcal{I} \cup \mathcal{I}_2{\downarrow_{(n-1)}} \left(s', r', \mathcal{I}_2\right)
\end{aligned}$$

as required.

□

Lemma 11 (Shift-Closure-2)**.** For all $\mathcal{I}_0, \mathcal{I}_1, \mathcal{I}_2, \mathcal{I} \in \mathsf{AMod}$ and $s_1, r_1, s_0, r_0 \in \mathsf{LState}$

$$\begin{aligned}
&\mathcal{I}{\downarrow} (s_1, r_1, \mathcal{I}_1) \land \mathcal{I}_1 \sqsubseteq^{\{s_1\}} \mathcal{I}_2 \land \\
&\mathcal{I}{\downarrow} (s_0, r_0, \mathcal{I}_0) \land s_1 \circ r_1 = s_0 \circ r_0 \implies \\
&\qquad\qquad\qquad \mathcal{I} \cup \mathcal{I}_2{\downarrow} (s_0, r_0, \mathcal{I}_0)
\end{aligned}$$

Proof. Pick an arbitrary $\mathcal{I}_0, \mathcal{I}_1, \mathcal{I}_2, \mathcal{I} \in \mathsf{AMod}$ and $s_1, r_1, s_0, r_0 \in \mathsf{LState}$ such that

$$\mathcal{I}{\downarrow} (s_1, r_1, \mathcal{I}_1) \tag{A.58}$$

$$\mathcal{I}_1 \sqsubseteq^{\{s_1\}} \mathcal{I}_2 \tag{A.59}$$

$$\mathcal{I}{\downarrow} (s_0, r_0, \mathcal{I}_0) \tag{A.60}$$

$$s_1 \circ r_1 = s_0 \circ r_0 \tag{A.61}$$

From the definition of $\downarrow$, it then suffices to show

$$\mathcal{I}_0 \subseteq \mathcal{I} \cup \mathcal{I}_2 \tag{A.62}$$

$$\forall n \in \mathbb{N}.\, \mathcal{I} \cup \mathcal{I}_2 {\downarrow_n} (s_0, r_0, \mathcal{I}_0) \tag{A.63}$$

RTS. (A.62)
From (A.60) and the definition of $\downarrow$ we have $\mathcal{I}_0 \subseteq \mathcal{I}$ and consequently $\mathcal{I}_0 \subseteq \mathcal{I} \cup \mathcal{I}_2$ as required.

RTS. (A.63)
Rather than proving (A.63) directly, we first establish the following.

$$\begin{aligned}
&\forall n \in \mathbb{N}.\, \forall s_1, r_1, s_0, r_0 \in \mathsf{LState}.\\
&\quad \mathcal{I}{\downarrow_n} (s_1, r_1, \mathcal{I}_1) \land \mathcal{I}_1 \sqsubseteq^{\{s_1\}} \mathcal{I}_2 \land\\
&\quad \mathcal{I}{\downarrow_n} (s_0, r_0, \mathcal{I}_0) \land s_1 \circ r_1 = s_0 \circ r_0 \implies\\
&\qquad\qquad\qquad \mathcal{I} \cup \mathcal{I}_2 {\downarrow_n} (s_0, r_0, \mathcal{I}_0)
\end{aligned} \tag{A.64}$$

We can then despatch (A.63) from (A.58)-(A.61) and (A.64); since for an arbitrary $n \in \mathbb{N}$, from (A.58), (A.60) and the definition of $\downarrow$ we have $\mathcal{I}{\downarrow_n} (s_1, r_1, \mathcal{I}_1) \land \mathcal{I}{\downarrow_n} (s_0, r_0, \mathcal{I}_0)$ and consequently from (A.59), (A.61) and (A.64) we derive $\mathcal{I} \cup \mathcal{I}_2 {\downarrow_n} (s_0, r_0, \mathcal{I}_0)$ as required.

RTS. (A.64)
We proceed by induction on the number of steps n.

Base case $n = 0$
Pick an arbitrary $s_1, r_1, s_0, r_0 \in \mathsf{LState}$. We are then required to show $\mathcal{I} \cup \mathcal{I}_2 {\downarrow_0} (s_0, r_0, \mathcal{I}_0)$ which follows trivially from the definition of $\downarrow_0$.

Inductive Case
Pick an arbitrary $n \in \mathbb{N}$ and $s_1, r_1, s_0, r_0 \in \mathsf{LState}$ such that:

$$\mathcal{I}{\downarrow_n} (s_1, r_1, \mathcal{I}_1) \tag{A.65}$$

$$\mathcal{I}_1 \sqsubseteq^{\{s_1\}} \mathcal{I}_2 \tag{A.66}$$

$$\mathcal{I}{\downarrow_n} (s_0, r_0, \mathcal{I}_0) \tag{A.67}$$

$$s_1 \circ r_1 = s_0 \circ r_0 \tag{A.68}$$

$$\begin{aligned} &\forall s'_1, r'_1, s'_0, r'_0 \in \mathsf{LState}. \\ &\quad \mathcal{I}{\downarrow_{(n-1)}} (s'_1, r'_1, \mathcal{I}_1) \wedge \mathcal{I}_1 \sqsubseteq^{\{s'_1\}} \mathcal{I}_2 \wedge \\ &\quad \mathcal{I}{\downarrow_{(n-1)}} (s'_0, r'_0, \mathcal{I}_0) \wedge s'_1 \circ r'_1 = s'_0 \circ r'_0 \implies \\ &\qquad\qquad\qquad\qquad \mathcal{I} \cup \mathcal{I}_2{\downarrow_{(n-1)}} (s'_0, r'_0, \mathcal{I}_0) \end{aligned} \tag{I.H}$$

RTS.

$$\begin{aligned} &\forall \kappa. \forall a \in (\mathcal{I} \cup \mathcal{I}_2)(\kappa). \, \mathit{potential}(a, s_0 \circ r_0) \Rightarrow \\ &\quad (\mathit{reflected}(a, s_0 \circ r_0, \mathcal{I}_0(\kappa)) \vee \neg \mathit{visible}(a, s_0)\ ) \wedge \\ &\quad \forall (s', r') \in a[s_0, r_0]. \, \mathcal{I} \cup \mathcal{I}_2{\downarrow_{(n-1)}} (s', r', \mathcal{I}_0) \end{aligned} \tag{A.69}$$

Pick an arbitrary κ and $a = (p, q, c) \in (\mathcal{I} \cup \mathcal{I}_2)(\kappa)$ such that

$$\mathit{potential}(a, s_0 \circ r_0) \tag{A.70}$$

There are two cases to consider:

Case 1. $a \in \mathcal{I}(\kappa)$
Then from (A.67) and assumption of case 1. we have:

$$\begin{aligned} &(\mathit{reflected}(a, s_0 \circ r_0, \mathcal{I}_0(\kappa)) \vee \neg \mathit{visible}(a, s_0)\ ) \wedge \\ &\forall (s', r') \in a[s_0, r_0]. \, \mathcal{I}{\downarrow_{(n-1)}} (s', r', \mathcal{I}_0) \end{aligned} \tag{A.71}$$

Pick an arbitrary (s', r') such that

$$(s', r') \in a[s_0, r_0] \tag{A.72}$$

Then from (A.71) and (A.72) we have:

$$\mathcal{I}{\downarrow_{(n-1)}} (s', r', \mathcal{I}_0) \tag{A.73}$$

From (A.72) and the definition of $a[s_0, r_0]$, we know that $p \leq s_0 \circ r_0$ and consequently from (A.68) we have $p \leq s_1 \circ r_1$. Thus from Lemma 27 we know there exists $p_s, p_r \in \mathsf{LState}$ such that :

$$p = p_s \circ p_r \wedge p_s \leq s_1 \wedge p_r \leq r_1$$

From the definition of $a[s_1, r_1]$ we then have

$$(p_s = \mathbf{0}_{\mathsf{L}} \wedge (s_1, q \circ (r_1 - p_r)) \in a[s_1, r_1])$$

$$\vee(p_s > \mathbf{0}_{\mathsf{L}} \wedge (q \circ (s_1 - p_s), r_1 - p_r) \in a[s_1, r_1])$$

That is, there exists $s'', r'' \in \mathsf{LState}$ such that

$$(s'', r'') \in a[s_1, r_1] \tag{A.74}$$

From (A.66), (A.74) and Lemma 9 we have:

$$\mathcal{I}_1 \sqsubseteq^{\{s''\}} \mathcal{I}_2 \tag{A.75}$$

From (A.68), (A.72), (A.74) and Lemma 12 we have:

$$s' \circ r' = s'' \circ r'' \tag{A.76}$$

From the assumption of case 1, (A.74) and (A.65) we have:

$$\begin{array}{l}(\mathit{reflected}(a, s_1 \circ r_1, \mathcal{I}_1(\kappa)) \vee \neg \mathit{visible}(a, s_1)\ ) \wedge \\ \forall (s', r') \in a[s_1, r_1].\, \mathcal{I}\!\downarrow_{(n-1)} (s', r', \mathcal{I}_1)\end{array}$$

and thus from (A.74) we have:

$$\mathcal{I}\!\downarrow_{(n-1)} \left(s'', r'', \mathcal{I}_1\right) \tag{A.77}$$

Consequently, from (A.73), (A.75), (A.76), (A.77) and (I.H) we have:

$$\mathcal{I} \cup \mathcal{I}_2\!\downarrow_{(n-1)} \left(s', r', \mathcal{I}_0\right)$$

as required.

Case 2. $a \in \mathcal{I}_2(\kappa)$
From the assumption of the case and (A.66) we have

$$\mathit{reflected}(a, s_1, \mathcal{I}_1(\kappa)) \tag{A.78}$$

From (A.68) and (A.70) we have $\mathit{potential}(a, s_1 \circ r_1)$ and consequently from the definition of *potential* we know there exists $l \in \mathsf{LState}$ such that $p \circ c \leq s_1 \circ r_1 \circ l$. Thus from (A.78) and the definition of *reflected* we have:

$$\exists a', c'.\, a' = (p, q, c') \wedge a' \in \mathcal{I}_1(\kappa) \wedge p \circ c' \leq s_1 \circ r_1 \circ l \tag{A.79}$$

From (A.68) and (A.70) we have $\mathit{potential}(a, s_1 \circ r_1)$. Consequently, from (A.79) and the definition of *potential* we have:

$$\mathit{potential}(a', s_1 \circ r_1) \tag{A.80}$$

Since $\mathcal{I}_1 \subseteq \mathcal{I}$, we know $a' \in \mathcal{I}(\kappa)$. Consequently, from (A.79), (A.80), (A.67), (A.68) and the definition of $a[s_0, r_0]$ we have

$$\forall (s', r') \in a[s_0, r_0].\, \mathcal{I}{\downarrow_{(n-1)}} (s', r', \mathcal{I}_0) \tag{A.81}$$

Similarly, from (A.79), (A.80) (A.65) and the definition of $a[s_1, r_1]$ we have

$$\forall (s', r') \in a[s_1, r_1].\, \mathcal{I}{\downarrow_{(n-1)}} (s', r', \mathcal{I}_1) \tag{A.82}$$

Pick an arbitrary (s', r') such that

$$(s', r') \in a[s_0, r_0] \tag{A.83}$$

Then from (A.81) we have:

$$\mathcal{I}{\downarrow_{(n-1)}} \left(s', r', \mathcal{I}_0\right) \tag{A.84}$$

From (A.83) and the definition of $a[s_0, r_0]$, we know that $p \leq s_0 \circ r_0$ and consequently from (A.68) we have $p \leq s_1 \circ r_1$. Thus from Lemma 27 we know there exists $p_s, p_r \in \mathsf{LState}$ such that :

$$p = p_s \circ p_r \wedge p_s \leq s_1 \wedge p_r \leq r_1$$

From the definition of $a[s_1, r_1]$ we then have

$$\begin{gathered}(p_s = \mathbf{0}_\mathsf{L} \wedge (s_1, q \circ (r_1 - p_r)) \in a'[s_1, r_1]) \\ \vee (p_s > \mathbf{0}_\mathsf{L} \wedge (q \circ (s_1 - p_s), r_1 - p_r) \in a'[s_1, r_1])\end{gathered}$$

That is, there exists $s'', r'' \in \mathsf{LState}$ such that

$$(s'', r'') \in a[s_1, r_1] \tag{A.85}$$

Thus from (A.82) we have:

$$\mathcal{I}{\downarrow_{(n-1)}} \left(s'', r'', \mathcal{I}_1\right) \tag{A.86}$$

From (A.83), (A.85), (A.68) and the definition of $a[s_1, r_1]$ we have:

$$s' \circ r' = s'' \circ r'' \tag{A.87}$$

From (A.66) and (A.83) and Lemma 9 we have:

$$\mathcal{I}_1 \sqsubseteq^{\{s'\}} \mathcal{I}_2 \tag{A.88}$$

From (A.84), (A.86), (A.87), (A.88) and (I.H) we have:

$$\mathcal{I} \cup \mathcal{I}_2 \downarrow_{(n-1)} \left(s', r', \mathcal{I}_0\right)$$

and thus from (A.83)

$$\forall (s', r') \in a[s_0, r_0].\, \mathcal{I} \cup \mathcal{I}_2 \downarrow_{(n-1)} \left(s', r', \mathcal{I}_0\right) \tag{A.89}$$

Since either $\mathit{visible}(a, s_0)$ or $\neg \mathit{visible}(a, s_0)$, there are two cases to consider:

Case 2.1. $\neg \mathit{visible}(a, s_0)$
From (A.89) and the assumption of case 2.1. we then have

$$\begin{array}{l} \neg \mathit{visible}(a, s_0) \wedge \\ \forall (s', r') \in a[s_0, r_0].\, \mathcal{I} \cup \mathcal{I}_2 \downarrow_{(n-1)} (s', r', \mathcal{I}_0) \end{array}$$

as required.

Case 2.2. $\mathit{visible}(a, s_0)$
Pick an arbitrary l such that

$$p \circ c \leq s_0 \circ r_0 \circ l \tag{A.90}$$

From (A.90) and (A.68) we then know $p \circ c \leq s_1 \circ r_1 \circ l$; from (A.78) and the definition of *reflected* we thus have

$$\exists a_1, c_1.\, a_1 = (p, q, c_1) \wedge a_1 \in \mathcal{I}_1(\kappa) \wedge p \circ c_1 \leq s_1 \circ r_1 \circ l \tag{A.91}$$

Since $\mathcal{I}_1 \subseteq \mathcal{I}$, from (A.91) we know $a_1 \in \mathcal{I}(\kappa)$. Consequently, from (A.67), (A.70), the assumption of case 2.2. and the definition of *reflected* we then have:

$$\exists a_0, c_0.\, a_0 = (p, q, c_0) \wedge a_0 \in \mathcal{I}_0(\kappa) \wedge p \circ c_0 \leq s_1 \circ r_1 \circ l \tag{A.92}$$

Consequently, from (A.91), (A.92) and (A.68) we have:

$$\exists a_0, c_0.\, a_0 = (p, q, c_0) \wedge a_0 \in \mathcal{I}_0(\kappa) \wedge p \circ c_0 \leq s_0 \circ r_0 \circ l \tag{A.93}$$

Thus from (A.90), (A.93) and by definition of *reflected* we have:

$$\mathit{reflected}(a, s_0 \circ r_0, \mathcal{I}_0(\kappa)) \tag{A.94}$$

Finally, from (A.89) and (A.94) we have:

$$\begin{array}{l} \mathit{reflected}(a, s_0 \circ r_0, \mathcal{I}_0(\kappa)) \wedge \\ \forall (s', r') \in a[s_0, r_0].\, \mathcal{I} \cup \mathcal{I}_2 \downarrow_{(n-1)} \left(s', r', \mathcal{I}_0\right) \end{array}$$

as required.

□

Lemma 12 (action-application)**.** For all $a \in \mathsf{LState} \times \mathsf{LState} \times \mathsf{LState}$ and s_1 , r_1, s_2, r_2, s'_1, r'_1, s'_2, $r'_2 \in \mathsf{LState}$,

$$s_1 \circ r_1 = s_2 \circ r_2 \wedge (s'_1, r'_1) \in a[s_1, r_1] \wedge (s'_2, r'_2) \in a[s_2, r_2] \implies s'_1 \circ r'_1 = s'_2 \circ r'_2$$

Proof. Take arbitrary $a \in \mathsf{LState} \times \mathsf{LState} \times \mathsf{LState}$ and s_1 , r_1, s_2, r_2, s'_1, r'_1, s'_2, $r'_2 \in \mathsf{LState}$ such that

$$s_1 \circ r_1 = s_2 \circ r_2 \tag{A.95}$$
$$(s'_1, r'_1) \in a[s_1, r_1] \tag{A.96}$$
$$(s'_2, r'_2) \in a[s_2, r_2] \tag{A.97}$$

Then from (A.96), and the definitions of $a[s_1, r_1]$ and $a[s_1 \circ r_1]$ we have:

$$a[s_1 \circ r_1] = s'_1 \circ r'_1 \tag{A.98}$$

Similarly, from (A.97) we have:

$$a[s_2 \circ r_2] = s'_2 \circ r'_2 \tag{A.99}$$

Finally, from (A.95), (A.98) and (A.99) we have:

$$s'_1 \circ r'_1 = s'_2 \circ r'_2$$

as required. □

Lemma 13 (Extend-Closure-1)**.** For all $\mathcal{I}, \mathcal{I}_e, \mathcal{I}_0 \in \mathsf{AMod}$ such that $\forall \kappa \in \mathsf{dom}(\mathcal{I}_0) \,.\, \mathcal{I}_0(\kappa) = \emptyset$; and for all $g, s_e \in \mathsf{LState}$ and $\mathcal{F}, \mathcal{F}_e \in \mathcal{P}(\mathsf{LState})$,

$$g \in \mathcal{F} \wedge \mathcal{F} \blacktriangleright \mathcal{I} \wedge s_e \in \mathcal{F}_e \wedge \mathcal{F}_e \blacktriangleright \mathcal{I}_e \cup \mathcal{I}_0 \implies \mathcal{I} \cup \mathcal{I}_e \cup \mathcal{I}_0 \downarrow (s_e, g, \mathcal{I}_e)$$

Proof. Pick an arbitrary $\mathcal{I}, \mathcal{I}_e, \mathcal{I}_0 \in \mathsf{AMod}$, $g, s_e \in \mathsf{LState}$ and $\mathcal{F}, \mathcal{F}_e \in \mathcal{P}(\mathsf{LState})$ such that

$$\forall \kappa \in \mathsf{dom}(\mathcal{I}_0) \,.\, \mathcal{I}_0(\kappa) = \emptyset \tag{A.100}$$
$$g \in \mathcal{F} \wedge s_e \in \mathcal{F}_e \tag{A.101}$$
$$\mathcal{F} \blacktriangleright \mathcal{I} \wedge \mathcal{F}_e \blacktriangleright \mathcal{I}_e \cup \mathcal{I}_0 \tag{A.102}$$

From the definition of $\downarrow$, it then suffices to show

$$\mathcal{I}_e \subseteq \mathcal{I} \cup \mathcal{I}_e \cup \mathcal{I}_0 \tag{A.103}$$
$$\forall n \in \mathbb{N}.\, \mathcal{I} \cup \mathcal{I}_e \downarrow_n (s_e, g, \mathcal{I}_e) \tag{A.104}$$

RTS. (A.103)
This holds trivially from the definition of $\mathcal{I} \cup \mathcal{I}_e$.

RTS. (A.104)
Rather than proving (A.104) directly, we first establish the following.

$$\begin{aligned} &\forall n \in \mathbb{N}.\, \forall g, s_e \in \mathsf{LState}. \\ &\quad g \in \mathcal{F} \wedge s_e \in \mathcal{F}_e \implies \mathcal{I} \cup \mathcal{I}_e \cup \mathcal{I}_0 \downarrow_n (s_e, g, \mathcal{I}_e) \end{aligned} \tag{A.105}$$

We can then despatch (A.104) from (A.101) and (A.105); since for an arbitrary $n \in \mathbb{N}$, from (A.101) and (A.105) we have $\mathcal{I} \cup \mathcal{I}_e \cup \mathcal{I}_0 \downarrow_n (s_e, g, \mathcal{I}_e)$ as required.

RTS. (A.105)
We proceed by induction on the number of steps n.

Base case $n = 0$
Pick an arbitrary $g, s_e \in \mathsf{LState}$. We are then required to show $\mathcal{I} \cup \mathcal{I}_e \cup \mathcal{I}_0 \downarrow_0 (s_e, g, \mathcal{I}_e)$ which follows trivially from the definition of $\downarrow_0$.

Inductive case
Pick an arbitrary $n \in \mathbb{N}$ and $g, s_e \in \mathsf{LState}$ such that

$$g \in \mathcal{F} \tag{A.106}$$

$$s_e \in \mathcal{F}_e \tag{A.107}$$
$$\forall g', s'_e.\, g' \in \mathcal{F} \wedge s'_e \in \mathcal{F}_e \implies \mathcal{I} \cup \mathcal{I}_e \cup \mathcal{I}_0 {\downarrow_{(n-1)}} \left(s'_e, g', \mathcal{I}_e\right) \tag{I.H}$$

RTS.

$$\begin{array}{l} \forall \kappa.\, \forall a \in (\mathcal{I} \cup \mathcal{I}_e \cup \mathcal{I}_0)\,(\kappa).\, \mathit{potential}(a, s_e \circ g) \Rightarrow \\ \quad (\mathit{reflected}(a, s_e \circ g, \mathcal{I}_e(\kappa)) \vee \neg \mathit{visible}(a, s_e)\ ) \wedge \\ \quad \forall (s', r') \in a[s_e, g].\, \mathcal{I} \cup \mathcal{I}_e \cup \mathcal{I}_0 {\downarrow_{(n-1)}} (s', r', \mathcal{I}_e) \end{array} \tag{A.108}$$

RTS. A.108
Pick an arbitrary κ, $a = (p, q, c) \in (\mathcal{I} \cup \mathcal{I}_e \cup \mathcal{I}_0)\,(\kappa)$ and (s', r') such that

$$\mathit{potential}(a, s_e \circ g) \tag{A.109}$$
$$(s', r') \in a[s_e, g] \tag{A.110}$$

Since from (A.100) we know $\mathcal{I}_0(\kappa) = \emptyset$ we know $a \in \mathcal{I}_e(\kappa) \vee a \in \mathcal{I}(\kappa)$, and thus there are two cases to consider.

Case 1. $a \in \mathcal{I}(\kappa)$
From (A.109) and the definition of *potential* we have $s_e \circ g \sqcap p \circ c \neq \emptyset$ and consequently, $g \sqcap p \circ c \neq \emptyset$, from (A.106) we then have:

$$p \leq g \wedge p \perp s_e \tag{A.111}$$

From (A.109), (A.111) and the definitions of *potential* we have:

$$\mathit{potential}(a, g) \tag{A.112}$$

From (A.111) and the definition of *visible* we have:

$$\neg \mathit{visible}(a, s_e) \tag{A.113}$$

On the other hand, from (A.110), (A.111) and the definitions of $a[s_e, g]$, $a[g]$ and $\perp$, we know:

$$s' = s_e \tag{A.114}$$
$$a[g] = r' \tag{A.115}$$

Consequently, from (A.102), (A.106), (A.112), (A.115) and the definition of $\blacktriangleright$ we have:

$$r' \in \mathcal{F} \tag{A.116}$$

Finally, from (A.107), (A.116), (I.H) we have:

$$\mathcal{I} \cup \mathcal{I}_e \cup \mathcal{I}_0 {\downarrow_{(n-1)}} \left(s_e, r', \mathcal{I}_e\right) \tag{A.117}$$

and consequently from (A.114) and (A.110)-(A.117) we have

$$\begin{array}{l} \neg \mathit{visible}(a, s_e) \wedge \\ \forall (s', r') \in a[s_e, g].\, \mathcal{I} \cup \mathcal{I}_e \cup \mathcal{I}_0 {\downarrow_{(n-1)}} \left(s_e, r', \mathcal{I}_e\right) \end{array}$$

as required.

Case 2. $a \in \mathcal{I}_e(\kappa)$
From the assumption of the case and the definition of *reflected* we trivially have

$$\mathit{reflected}(a, s_e \circ g, \mathcal{I}_e(\kappa)) \tag{A.118}$$

Since from (A.109) and the definition of *potential* we have $s_e \circ g \sqcap p \circ c \neq \emptyset$ and consequently, $s_e \sqcap p \circ c \neq \emptyset$, from (A.107) we have:

$$p \leq s_e \wedge p \perp g \tag{A.119}$$

From (A.109), (A.119) and the definition of *potential* we have:

$$\mathit{potential}(a, s_e) \tag{A.120}$$

On the other hand, from (A.110), (A.119) and the definitions of $a[s_e, g]$ and $\perp$, we have:

$$r' = g \tag{A.121}$$
$$a[s_e] = s' \tag{A.122}$$

Consequently, from (A.102), (A.107), (A.120), (A.122) and the definition of $\blacktriangleright$ we have:

$$s' \in \mathcal{F}_e \tag{A.123}$$

Finally, from (A.106), (A.123) and (I.H) we have:

$$\mathcal{I} \cup \mathcal{I}_e \cup \mathcal{I}_0 {\downarrow_{(n-1)}} \left(s', g, \mathcal{I}_e\right)$$

and consequently from (A.121)

$$\mathcal{I} \cup \mathcal{I}_e \cup \mathcal{I}_0 {\downarrow_{(n-1)}} \left(s', r', \mathcal{I}_e\right) \tag{A.124}$$

Thus from (A.110) and (A.118)-(A.124) we have:

$$\begin{aligned}&\mathit{reflected}(a, s_e \circ g, \mathcal{I}_e(\kappa)) \land \\ &\forall (s', r') \in a[s_e, g].\, \mathcal{I} \cup \mathcal{I}_e \cup \mathcal{I}_0 {\downarrow_{(n-1)}} \left(s', r', \mathcal{I}_e\right)\end{aligned}$$

as required.

$\square$

Lemma 14 (Extend-closure-2)**.** For all $\mathcal{I}_0, \mathcal{I}, \mathcal{I}_e \in \mathsf{AMod}$, $s, g, s_e \in \mathsf{LState}$ and $\mathcal{F}, \mathcal{F}_e \in \mathcal{P}(\mathsf{LState})$

$$\begin{aligned}&g \in \mathcal{F} \land \mathcal{F} \blacktriangleright \mathcal{I} \land s_e \in \mathcal{F}_e \land \mathcal{F}_e \blacktriangleright \mathcal{I}_e \land \mathcal{I}{\downarrow} (s, g - s, \mathcal{I}_0) \\ &\quad \implies \mathcal{I} \cup \mathcal{I}_e{\downarrow} (s, (g - s) \circ s_e, \mathcal{I}_0)\end{aligned}$$

Proof. Pick an arbitrary $\mathcal{I}_0, \mathcal{I}, \mathcal{I}_e \in \mathsf{AMod}$, $s, g, s_e \in \mathsf{LState}$ and $\mathcal{F}, \mathcal{F}_e \in \mathcal{P}(\mathsf{LState})$ such that

$$g \in \mathcal{F} \land s_e \in \mathcal{F}_e \tag{A.125}$$
$$\mathcal{F} \blacktriangleright \mathcal{I} \land \mathcal{F}_e \blacktriangleright \mathcal{I}e \tag{A.126}$$
$$\mathcal{I}{\downarrow} (s, g - s, \mathcal{I}_0) \tag{A.127}$$

From the definition of $\downarrow$, it then suffices to show

$$\mathcal{I}_0 \subseteq \mathcal{I} \cup \mathcal{I}_2 \tag{A.128}$$
$$\forall n \in \mathbb{N}.\, \mathcal{I} \cup \mathcal{I}_e{\downarrow_n} (s, (g - s) \circ s_e, \mathcal{I}_0) \tag{A.129}$$

RTS. (A.128)
Since from (A.127) and the definition of $\downarrow$ we have $\mathcal{I}_0 \subseteq \mathcal{I}$, we can consequently derive $\mathcal{I}_0 \subseteq \mathcal{I} \cup \mathcal{I}_e$ as required.

RTS. (A.129)
Rather than proving (A.129) directly, we first establish the following.

$$\begin{aligned}&\forall n \in \mathbb{N}.\, \forall s, g, s_e \in \mathsf{LState}. \\ &\quad g \in \mathcal{F} \land s_e \in \mathcal{F}_e \land \mathcal{I}{\downarrow_n} (s, g - s, \mathcal{I}_0) \implies \\ &\qquad \mathcal{I} \cup \mathcal{I}_e{\downarrow_n} (s, (g - s) \circ s_e, \mathcal{I}_0)\end{aligned} \tag{A.130}$$

We can then despatch (A.129) from (A.125)-(A.127) and (A.130); since for an arbitrary $n \in \mathbb{N}$, from (A.127) and the definition of $\downarrow$ we have $\mathcal{I} \downarrow_n (s, g - s, \mathcal{I}_0)$ and consequently from (A.125) and (A.130) we derive

$\mathcal{I} \cup \mathcal{I}_e {\downarrow_n} (s, (g - s) \circ s_e, \mathcal{I}_0)$ as required.

RTS. (A.130)
We proceed by induction on the number of steps n.

Base case $n = 0$
Pick an arbitrary $s, g, s_e \in \mathsf{LState}$. We are then required to show $\mathcal{I} \cup \mathcal{I}_e {\downarrow_0}$ $(s, (g - s) \circ s_e, \mathcal{I}_0)$ which follows trivially from the definition of ${\downarrow_0}$.

Inductive case
Pick an arbitrary $n \in \mathbb{N}$ and $s, r, g, s_e \in \mathsf{LState}$ such that

$$g \in \mathcal{F} \tag{A.131}$$
$$s_e \in \mathcal{F}_e \tag{A.132}$$
$$g = s \circ r \tag{A.133}$$
$$\mathcal{I}{\downarrow_n} (s, r, \mathcal{I}_0) \tag{A.134}$$
$$\begin{aligned}&\forall s'', g'', s''_e.\, g'' \in \mathcal{F} \wedge s''_e \in \mathcal{F}_e \wedge \mathcal{I}{\downarrow_{(n-1)}} \left(s'', g'' - s'', \mathcal{I}_0\right) \\ &\qquad \implies \mathcal{I} \cup \mathcal{I}_e {\downarrow_{(n-1)}} \left(s'', (g'' - s'') \circ s''_e, \mathcal{I}_0\right)\end{aligned} \tag{I.H}$$

RTS.

$$\begin{aligned}&(\mathit{reflected}(a, g \circ s_e, \mathcal{I}_0(\kappa)) \vee \neg \mathit{visible}(a, s)\ ) \wedge \\ &\forall (s', r') \in a[s, r \circ s_e].\, \mathcal{I} \cup \mathcal{I}_e {\downarrow_{(n-1)}} (s', r', \mathcal{I}_0)\end{aligned}$$

Pick an arbitrary κ, $a = (p, q, c) \in (\mathcal{I} \cup \mathcal{I}_e)(\kappa)$ and (s', r') such that

$$\mathit{potential}(a, g \circ s_e) \tag{A.135}$$
$$(s', r') \in a[s, r \circ s_e] \tag{A.136}$$

Since either $a \in \mathcal{I}(\kappa)$ or $a \in \mathcal{I}_e(\kappa)$, there are two cases to consider:

Case 1. $a \in \mathcal{I}(\kappa)$
Since from (A.135) and the definition of *potential* we have $s_e \circ g \sqcap p \circ c \neq \emptyset$ and consequently, $g \sqcap p \circ c \neq \emptyset$, from (A.131) we have:

$$p \leq g \wedge p \perp s_e \tag{A.137}$$

From (A.135), (A.137) and the definition of *potential* we have:

$$\mathit{potential}(a, g) \tag{A.138}$$

On the other hand, from (A.136), (A.137) and the definitions of $a[s, r \circ s_e]$ and $\perp$, we know there exists r'':

$$r' = r'' \circ s_e \tag{A.139}$$
$$(s', r'') \in a[s, r] \tag{A.140}$$

From (A.140) and the definitions of $a[s, r]$ and $a[s \circ r]$, we know $s' \circ r'' = a[s \circ r]$. Consequently, from (A.126), (A.131), (A.138), the definition of $\blacktriangleright$ and since $g = s \circ r$ (A.133), we have:

$$s' \circ r'' \in \mathcal{F} \tag{A.141}$$

On the other hand, from (A.134), (A.138), (A.135) and (A.140) we have:

$$(\mathit{reflected}(a, s \circ r, \mathcal{I}_0(\kappa)) \vee \neg\mathit{visible}(a, s)) \wedge \tag{A.142}$$
$$\mathcal{I}{\downarrow_{(n-1)}} \left(s', r'', \mathcal{I}_0\right) \tag{A.143}$$

From (A.132), (A.141), (A.142) and (I.H) we have:

$$\mathcal{I} \cup \mathcal{I}_e{\downarrow_{(n-1)}} \left(s', r'' \circ s_e, \mathcal{I}_0\right)$$

and thus from (A.139)

$$\mathcal{I} \cup \mathcal{I}_e{\downarrow_{(n-1)}} \left(s', r', \mathcal{I}_0\right) \tag{A.144}$$

Consequently, from (A.136)-(A.144) we have:

$$\forall (s', r') \in a[s, r \circ s_e].\, \mathcal{I} \cup \mathcal{I}_e{\downarrow_{(n-1)}} \left(s', r', \mathcal{I}_0\right) \tag{A.145}$$

From (A.142) there are two cases to consider:
Case 1.1. $\neg\mathit{visible}(a, s)$
From (A.145) and the assumption of the case we have:

$$\neg\mathit{visible}(a, s) \wedge$$
$$\forall (s', r') \in a[s, r \circ s_e].\, \mathcal{I} \cup \mathcal{I}_e{\downarrow_{(n-1)}} \left(s', r', \mathcal{I}_0\right)$$

as required.

Case 1.2. $\mathit{reflected}(a, s \circ r, \mathcal{I}_0(\kappa)$
Pick an arbitrary $l \in \mathsf{LState}$ such that

$$p \circ c \leq g \circ s_e \circ l \tag{A.146}$$

From the assumption of the case, (A.133) and the definition of *reflected* we then have

$$\exists a', c'.\, a' = (p, q, c') \wedge a' \in \mathcal{I}_0(\kappa) \wedge p \circ c' \leq g \circ s_e \circ l \tag{A.147}$$

Thus from (A.146), (A.147) and the definition of *reflected* we have:

$$\mathit{reflected}(a, g \circ s_e, \mathcal{I}_0(\kappa)) \tag{A.148}$$

Thus from (A.145) and (A.148) we have:

$$\begin{aligned}&\mathit{reflected}(a, g \circ s_e, \mathcal{I}_0(\kappa)) \wedge \\ &\forall (s', r') \in a[s, r \circ s_e].\, \mathcal{I} \cup \mathcal{I}_e \downarrow_{(n-1)} \left(s', r', \mathcal{I}_0\right)\end{aligned}$$

as required.

Case 2. $a \in \mathcal{I}_e(\kappa)$
Since from (A.135) and the definition of *potential* we have $g \circ s_e \sqcap p \circ c \neq \emptyset$ and consequently, $s_e \sqcap p \circ c \neq \emptyset$, from (A.132) and the assumption of the case we have:

$$p \leq s_e \wedge p \perp g \tag{A.149}$$

and consequently from the definition of *visible* and (A.133) we have

$$\neg \mathit{visible}(a, s) \tag{A.150}$$

From (A.135), (A.149) and the definition of *potential* we have:

$$\mathit{potential}(a, s_e) \tag{A.151}$$

From (A.133), (A.149) and the definitions of $a[s, r \circ s_e]$ and $\perp$, we know there exists s'_e such that:

$$s' = s \wedge r' = r \circ s'_e \tag{A.152}$$

$$a[s_e] = s'_e \tag{A.153}$$

Consequently, from (A.126), (A.132), (A.151), (A.153) and the definition of $\blacktriangleright$ we have:

$$s'_e \in \mathcal{F}_e \tag{A.154}$$

From (A.133), (A.152) and Lemma 15 we have:

$$\mathcal{I} \downarrow_{(n-1)} \left(s', r, \mathcal{I}_0\right) \tag{A.155}$$

From (A.131), (A.154), (A.155), (I.H) we have:

$$\mathcal{I} \cup \mathcal{I}_e{\downarrow}_{(n-1)} \left(s', r \circ s'_e, \mathcal{I}_0\right)$$

and thus from (A.152)

$$\mathcal{I} \cup \mathcal{I}_e{\downarrow}_{(n-1)} \left(s', r', \mathcal{I}_0\right) \tag{A.156}$$

Finally, from (A.136), (A.150) and (A.156) we have:

$$\begin{array}{l} \neg \mathit{visible}(a, s) \wedge \\ \forall (s', r') \in a[s, r \circ s_e].\, \mathcal{I} \cup \mathcal{I}_e{\downarrow}_{(n-1)} \left(s', r', \mathcal{I}_0\right) \end{array}$$

as required.

□

Lemma 15. For all $\mathcal{I}, \mathcal{I}' \in \mathsf{AMod}$ and $n \in \mathbb{N}^+$

$$\forall s, r \in \mathsf{LState}.\, \mathcal{I}{\downarrow_n} \left(s, r, \mathcal{I}'\right) \implies \mathcal{I}{\downarrow_{(n-1)}} \left(s, r, \mathcal{I}'\right)$$

Proof. Pick an arbitrary $s, r \in \mathsf{LState}$ and proceed by induction on number of steps n.

Base case: n = 1
Pick an arbitrary $s, r \in \mathsf{LState}$. We are then required to show $\mathcal{I}{\downarrow_0} (s, r, \mathcal{I}')$ which trivially follows from the definition of $\downarrow_0$.

Inductive case
Pick an arbitrary $s, r \in \mathsf{LState}$ such that

$$\mathcal{I}{\downarrow_{(n+1)}} \left(s, r, \mathcal{I}'\right) \tag{A.157}$$
$$\forall s', r' \in \mathsf{LState}.\, \mathcal{I}{\downarrow_n} \left(s', r', \mathcal{I}'\right) \implies \mathcal{I}{\downarrow_{(n-1)}} \left(s', r', \mathcal{I}'\right) \tag{I.H}$$

RTS.

$$\begin{array}{l} \forall \kappa.\, \forall a \in \mathcal{I}(\kappa).\, \mathit{potential}(a, s \circ r) \Rightarrow \\ (\mathit{reflected}(a, s \circ r, \mathcal{I}'(\kappa)) \vee \neg \mathit{visible}(a, s)\ ) \wedge \\ \forall (s', r') \in a[s, r].\, \mathcal{I}{\downarrow_{(n-1)}} (s', r', \mathcal{I}') \end{array}$$

Pick an arbitrary κ and $a \in \mathcal{I}(\kappa)$ such that

$$\mathit{potential}(a, s \circ r) \tag{A.158}$$

Then from (A.157) we have:

$$\begin{array}{l} (\mathit{reflected}(a, s \circ r, \mathcal{I}'(\kappa)) \vee \neg \mathit{visible}(a, s)\ ) \wedge \\ \forall (s', r') \in a[s, r].\, \mathcal{I}{\downarrow_n} (s', r', \mathcal{I}') \end{array}$$

and consequently from (I.H)

$$\begin{array}{l} (\mathit{reflected}(a, s \circ r, \mathcal{I}'(\kappa)) \vee \neg \mathit{visible}(a, s)\ ) \wedge \\ \forall (s', r') \in a[s, r].\, \mathcal{I}{\downarrow_{(n-1)}} (s', r', \mathcal{I}') \end{array}$$

as required.

□

Lemma 16 (Sequential command soundness)**.** For all $\mathbb{S} \in \mathsf{Seqs}$, $(M_1, \mathbb{S}, M_2) \in \mathrm{Ax}_{\mathsf{S}}$ and $m \in \mathbb{M}$:

$$\|\mathbb{S}\|_{\mathsf{S}} \left(\lfloor M_1 \bullet_{\mathbb{M}} \{m\} \rfloor_{\mathbb{M}}\right) \subseteq \lfloor M_2 \bullet_{\mathbb{M}} \{m\} \rfloor_{\mathbb{M}}$$

Proof. By induction over the structure of $\mathbb{S}$. Pick an arbitrary $m \in \mathbb{M}$.

Case $\mathbb{E}$
This follows immediately from parameter 10.

Case skip
RTS.

$$\|\mathbb{S}\|_{\mathsf{S}} \left(\lfloor M \bullet_{\mathbb{M}} \{m\} \rfloor_{\mathbb{M}}\right) \subseteq \lfloor M \bullet_{\mathbb{M}} \{m\} \rfloor_{\mathbb{M}}$$

Proof.

$$\begin{array}{rcl} \|\mathbb{S}\|_{\mathsf{S}} \left(\lfloor M \bullet_{\mathbb{M}} \{m\} \rfloor_{\mathbb{M}}\right) & = & \lfloor M \bullet_{\mathbb{M}} \{m\} \rfloor_{\mathbb{M}} \\ & \subseteq & \lfloor M \bullet_{\mathbb{M}} \{m\} \rfloor_{\mathbb{M}} \end{array}$$

as required.

Case $\mathbb{S}_1; \mathbb{S}_2$
RTS.

$$\|\mathbb{S}_1; \mathbb{S}_2\|_{\mathsf{S}} \left(\lfloor M \bullet_{\mathbb{M}} \{m\} \rfloor_{\mathbb{M}}\right) \subseteq \lfloor M' \bullet_{\mathbb{M}} \{m\} \rfloor_{\mathbb{M}}$$

where $(M, \mathbb{S}_1, M''), (M'', \mathbb{S}_2, M') \in \mathrm{Ax}_{\mathsf{S}}$.

Proof.

$$\begin{array}{rcl} \|\mathbb{S}_1; \mathbb{S}_2\|_{\mathsf{S}} \left(\lfloor M \bullet_{\mathbb{M}} \{m\} \rfloor_{\mathbb{M}}\right) & = & \|\mathbb{S}_2\|_{\mathsf{S}} \left(\|\mathbb{S}_1\|_{\mathsf{S}} \left(\lfloor M \bullet_{\mathbb{M}} \{m\} \rfloor_{\mathbb{M}}\right)\right) \\ \text{(I.H.)} & \subseteq & \|\mathbb{S}_2\|_{\mathsf{S}} \left(\lfloor M'' \bullet_{\mathbb{M}} \{m\} \rfloor_{\mathbb{M}}\right) \\ \text{(I.H.)} & \subseteq & \lfloor M' \bullet_{\mathbb{M}} \{m\} \rfloor_{\mathbb{M}} \end{array}$$

as required.

Case $\mathbb{S}_1 + \mathbb{S}_2$
RTS.

$$\|\mathbb{S}_1 + \mathbb{S}_2\|_{\mathsf{S}} \left(\lfloor M \bullet_{\mathbb{M}} \{m\} \rfloor_{\mathbb{M}}\right) \subseteq \lfloor M' \bullet_{\mathbb{M}} \{m\} \rfloor_{\mathbb{M}}$$

where $(M, \mathbb{S}_1, M'), (M, \mathbb{S}_2, M') \in \mathrm{Ax}_{\mathsf{S}}$.

Proof.

$$\begin{array}{rcl} \|\mathbb{S}_1 + \mathbb{S}_2\|_{\mathsf{S}} \left(\lfloor M \bullet_{\mathbb{M}} \{m\} \rfloor_{\mathbb{M}}\right) & = & \|\mathbb{S}_1\|_{\mathsf{S}} \left(\lfloor M \bullet_{\mathbb{M}} \{m\} \rfloor_{\mathbb{M}}\right) \cup \|\mathbb{S}_2\|_{\mathsf{S}} \left(\lfloor M \bullet_{\mathbb{M}} \{m\} \rfloor_{\mathbb{M}}\right) \\ \text{(I.H.)} & \subseteq & \lfloor M' \bullet_{\mathbb{M}} \{m\} \rfloor_{\mathbb{M}} \cup \lfloor M' \bullet_{\mathbb{M}} \{m\} \rfloor_{\mathbb{M}} \\ & \subseteq & \lfloor M' \bullet_{\mathbb{M}} \{m\} \rfloor_{\mathbb{M}} \end{array}$$

as required.

Case $\mathbb{S}^\star$
RTS.

$$\llbracket \mathbb{S}^\star \rrbracket_{\mathsf{S}} \left(\lfloor M \bullet_{\mathbb{M}} \{m\} \rfloor_{\mathbb{M}} \right) \subseteq \lfloor M \bullet_{\mathbb{M}} \{m\} \rfloor_{\mathbb{M}}$$

where $(M, \mathbb{S}, M) \in \mathrm{Ax}_{\mathsf{S}}$.

Proof.

$$\begin{array}{rcl} \llbracket \mathbb{S}^\star \rrbracket_{\mathsf{S}} \left(\lfloor M \bullet_{\mathbb{M}} \{m\} \rfloor_{\mathbb{M}} \right) & = & \llbracket \mathrm{skip} + \mathbb{S}; \mathbb{S}^\star \rrbracket_{\mathsf{S}} \left(\lfloor M \bullet_{\mathbb{M}} \{m\} \rfloor_{\mathbb{M}} \right) \\ & = & \llbracket \mathrm{skip} \rrbracket_{\mathsf{S}} \left(\lfloor M \bullet_{\mathbb{M}} \{m\} \rfloor_{\mathbb{M}} \right) \cup \llbracket \mathbb{S}; \mathbb{S}^\star \rrbracket_{\mathsf{S}} \left(\lfloor M \bullet_{\mathbb{M}} \{m\} \rfloor_{\mathbb{M}} \right) \\ \text{(I.H.)} & \subseteq & \lfloor M \bullet_{\mathbb{M}} \{m\} \rfloor_{\mathbb{M}} \cup \lfloor M \bullet_{\mathbb{M}} \{m\} \rfloor_{\mathbb{M}} \\ & \subseteq & \lfloor M \bullet_{\mathbb{M}} \{m\} \rfloor_{\mathbb{M}} \end{array}$$

as required.

□

Lemma 17. For all $w_1, w_2, w, w' = (l', g', \mathcal{I}') \in \mathsf{World}$,

$$w_1 \bullet w_2 = w \wedge (l', g', \mathcal{I}') \in G(w_1) \implies ((w_2)_{\mathsf{L}}, g', \mathcal{I}') \in R(w_2)$$

Proof. Pick an arbitrary w_1, w_2, w such that:

$$w_1 \bullet w_2 = w \tag{A.159}$$
$$(l', g', \mathcal{I}') \in G(w_1) \tag{A.160}$$

RTS.

$$((w_2)_{\mathsf{L}}, g', \mathcal{I}') \in R(w_2)$$

* From (A.160) and by definition of G we know:

$$(l', g', \mathcal{I}') \in (G^{\mathsf{u}} \cup G^{\mathsf{e}} \cup G^{\mathsf{s}})^{\star}(w_2) \tag{A.161}$$

From (A.159), (A.161) and by Lemmata 18, 19 and 20 we have:

$$((w_2)_{\mathsf{L}}, g', \mathcal{I}') \in (R^{\mathsf{u}} \cup R^{\mathsf{e}} \cup R^{\mathsf{s}})^{\star}(w_2)$$

and consequently

$$((w_2)_{\mathsf{L}}, g', \mathcal{I}') \in R(w_2)$$

as required. □

Lemma 18. For all $w_1, w_2, w, w' = (l', g', \mathcal{I}') \in \mathsf{World}$,

$$w_1 \bullet w_2 = w \wedge (l', g', \mathcal{I}') \in G^{\mathsf{u}}(w_1) \implies ((w_2)_{\mathsf{L}}, g', \mathcal{I}') \in R^{\mathsf{u}}(w_2)$$

where we write $R^{\mathsf{u}}(w)$ to denote $\{w' \mid (w, w') \in R^{\mathsf{u}}(w)\}$.

Proof. Pick an arbitrary $W_1 = (l_1, g_1, \mathcal{I}_1)$, $w_2 = (l_2, g_2, \mathcal{I}_2), w$ and $(l', g', \mathcal{I}')$ such that:

$$w_1 \bullet w_2 = w \tag{A.162}$$
$$(l', g', \mathcal{I}') \in G^{\mathsf{u}}(w_1) \tag{A.163}$$

RTS.

$$((w_2)_{\mathsf{L}}, g', \mathcal{I}') \in R^{\mathsf{u}}(w_2)$$

* From (A.162) we know:

$$g_1 = g_2 \tag{A.164}$$
$$\mathcal{I}_1 = \mathcal{I}_2 \tag{A.165}$$

By definition of $G^{\mathsf{u}}$and from (A.163) and (A.165) we know:

$$\mathcal{I}' = \mathcal{I}_1 = \mathcal{I}_2 \tag{A.166}$$
$$(l_1 \circ g_1)_{\mathsf{K}}{}^{\perp}_{\mathbb{K}} = \left((l' \circ g')_{\mathsf{M}}\right)^{\perp}_{\mathbb{K}} \tag{A.167}$$
$$g' = g_1 \vee \left(\begin{array}{l} \exists \kappa \leq (l_1)_{\mathsf{K}}.\, (g_1, g') \in \lceil \mathcal{I}_1 \rceil\, (\kappa) \wedge \\ ((l_1 \circ g_1)_{\mathsf{M}})^{\perp}_{\mathbb{M}} = ((l' \circ g')_{\mathsf{M}})^{\perp}_{\mathbb{M}} \end{array} \right)$$

There are two cases to consider:

Case 1. $g_1 = g'$
From (A.164) and the assumption of the case we know $g' = g_2$. Consequently, from (A.166) we have:

$$((w_2)_{\mathsf{L}}, g', \mathcal{I}') = (l_2, g_2, \mathcal{I}_2) \tag{A.168}$$

By definition of R^{u} and from (A.168) we can conclude:

$$((w_2)_{\mathsf{L}}, g', \mathcal{I}') \in R^{\mathsf{u}}(l_2, g_2, \mathcal{I}_2) \tag{A.169}$$

as required.

Case 2.

$$\exists \kappa \leq (l_1)_{\mathsf{K}}.\, (g_1, g') \in \lceil \mathcal{I}_1 \rceil\, (\kappa) \tag{A.170}$$
$$((l_1 \circ g_1)_{\mathsf{M}})^{\perp}_{\mathbb{M}} = \left((l' \circ g')_{\mathsf{M}}\right)^{\perp}_{\mathbb{M}} \tag{A.171}$$

From (A.162), (A.164) and (A.165) we know that

$$w = (l_1 \circ l_2, g_2, \mathcal{I}_2) \tag{A.172}$$

Since $\mathsf{wf}\,(w)$ (by definition of World) and from (A.164) we know:

$$(l_1 \circ l_2 \circ g_2)_{\mathsf{K}} = (l_1)_{\mathsf{K}} \bullet_{\mathbb{K}} (l_2)_{\mathsf{K}} \bullet_{\mathbb{K}} (g_2) = (l_1 \circ g_1)_{\mathsf{K}} \bullet_{\mathbb{K}} (l_2)_{\mathsf{K}} \text{ is defined} \tag{A.173}$$
$$(l_1 \circ l_2 \circ g_1)_{\mathsf{M}} = (l_1 \bullet_{\mathbb{M}} g_1)_{\mathsf{M}} \bullet_{\mathbb{M}} (l_2)_{\mathsf{M}} \text{ is defined} \tag{A.174}$$

Since $\kappa_1 \leq (l_1)_{\mathsf{K}}$ (A.170), from (A.173) and Lemma 28, we know:

$$\kappa \sharp \ (l_2)_{\mathsf{K}} \bullet_{\mathbb{K}} (g_2)_{\mathsf{K}} \tag{A.175}$$

From (A.164), (A.171) and (A.174) we know

$$(l' \circ g')_{\mathsf{M}} \bullet_{\mathbb{M}} (l_2)_{\mathsf{M}} = (l' \circ l_2 \circ g')_{\mathsf{M}} \quad \text{is defined} \tag{A.176}$$

From (A.167) and (A.173) we know

$$(l' \circ g')_{\mathsf{K}} \bullet_{\mathbb{K}} (l_2)_{\mathsf{K}} = (l' \circ l_2 \circ g')_{\mathsf{K}} \quad \text{is defined} \tag{A.177}$$

From (A.176) and (A.177) we know $l'_1 \circ l_2 \circ g'$ is defined and consequently:

$$l_2 \circ g' \quad \text{is defined} \tag{A.178}$$

From (A.165), (A.170), (A.175), (A.178) and by definition of R^{u}, we have:

$$((w_2)_{\mathsf{L}}, g', \mathcal{I}') = (l_2, g_2, \mathcal{I}_2) \in R^{\mathsf{u}}(l_2, s_2, \mathcal{I}_2)$$

as required. □

Lemma 19. For all $w_1, w_2, w, w' = (l', g', \mathcal{I}') \in \mathsf{World}$,

$$w_1 \bullet w_2 = w \wedge w' \in G^{\mathsf{e}}(w_1) \implies ((w_2)_{\mathsf{L}}, g', \mathcal{I}') \in R^{\mathsf{e}}(w_2)$$

Proof. Pick an arbitrary $w_1 = (l_1, g_1, \mathcal{I}_1)$, $w_2 = (l_2, g_2, \mathcal{I}_2), w$ and $w' = (l', g', \mathcal{I}')$ such that:

$$w_1 \bullet w_2 = w \tag{A.179}$$

$$w' \in G^{\mathsf{e}}(w_1) \tag{A.180}$$

RTS.

$$((w_2)_{\mathsf{L}}, g', \mathcal{I}') \in R^{\mathsf{e}}(w_2)$$

From (A.179) we know:

$$g_1 = g_2 \wedge \mathcal{I}_1 = \mathcal{I}_2 \tag{A.181}$$

By definition of G^{e} and from (A.180) and (A.181) we know there exists $l_3, l_4, g'' \in \mathsf{LState}$, $\kappa_1, \kappa_2 \in \mathbb{K}$ and $\mathcal{I}''$ such that

$$\begin{aligned} & l_1 = l_3 \circ l_4 \wedge l'_1 = l_3 \circ (\mathbf{0}_{\mathbb{M}}, \kappa_1) \\ & \wedge g'' = l_4 \circ (\mathbf{0}_{\mathbb{M}}, \kappa_2) \wedge g' = g_2 \circ g'' \\ & \wedge \kappa_1 \bullet_{\mathbb{K}} \kappa_2 \prec \mathsf{dom}\,(\mathcal{I}'') \wedge \kappa_1 \bullet_{\mathbb{K}} \kappa_2 \perp \mathsf{dom}\,(\mathcal{I}_2) \\ & \wedge \mathcal{I}' = \mathcal{I}_2 \cup \mathcal{I}'' \wedge \mathcal{I}'' \uparrow^{(g'')} (g_2, \mathcal{I}_2) \wedge g'' \circledcirc \mathcal{I}'' \end{aligned} \tag{A.182}$$

From (A.182) and the definition of R^{e} we have:

$$((w_2)_{\mathsf{L}}, g', \mathcal{I}') \in R^{\mathsf{e}}(w_2)$$

as required. $\square$

Lemma 20. For all $w_1, w_2, w, w' = (l', g', \mathcal{I}') \in \mathsf{World}$,

$$w_1 \bullet w_2 = w \wedge w' \in G^{\mathsf{s}}(w_1) \implies ((w_2)_{\mathsf{L}}, g', \mathcal{I}') \in R^{\mathsf{s}}(w_2)$$

Proof. Pick an arbitrary $w_1 = (l_1, g_1, \mathcal{I}_1)$, $w_2 = (l_2, g_2, \mathcal{I}_2), w$ and $w' = (l', g', \mathcal{I}')$ such that:

$$w_1 \bullet w_2 = w \tag{A.183}$$
$$w' \in G^{\mathsf{s}}(w_1) \tag{A.184}$$

RTS.

$$((w_2)_{\mathsf{L}}, g', \mathcal{I}') \in R^{\mathsf{s}}(w_2)$$

From (A.183) we know:

$$g_1 = g_2 \wedge \mathcal{I}_1 = \mathcal{I}_2 \tag{A.185}$$

By definition of $G^{\mathsf{s}}$and from (A.184), and (A.185) we know there exists $\mathcal{I}'' \in \mathsf{AMod}$ such that

$$g' = g_2 \wedge \mathcal{I}' = \mathcal{I}_2 \cup \mathcal{I}'' \wedge \mathcal{I}'' \uparrow^{(\mathsf{0_L})} (g_2, \mathcal{I}_2) \tag{A.186}$$

From (A.186) and the definition of R^{s} we have:

$$((w_2)_{\mathsf{L}}, g', \mathcal{I}') \in R^{\mathsf{s}}(w_2)$$

as required. $\square$

Lemma 21. Given any assertion $P \in \mathsf{Assn}$, the following is valid.

$$P \Leftrightarrow \mathsf{promote}\,(P)$$

Proof. **TO DO** □

Lemma 22. for all $P \in \mathsf{Assn}$, $\iota \in \mathsf{LEnv}$, $w, w' \in \mathsf{World}$,

$$w, \iota \vDash P \land (w, w') \in R^{\mathsf{e}} \implies w', \iota \vDash P$$

Proof. From the definition of R^{e}, it then suffices to show that for all $P \in \mathsf{Assn}$, $\iota \in \mathsf{LEnv}$, $l, g, g' \in \mathsf{LState}$ and $\mathcal{I}, \mathcal{I}' \in \mathsf{AMod}$:

$$(l, g, \mathcal{I}), \iota \vDash P \land \mathcal{I}' \uparrow^{(g')} (g, \mathcal{I}) \implies (l, g \circ g', \mathcal{I} \cup \mathcal{I}'), \iota \vDash P$$

We proceed by induction on the structure of assertion P.

Case $P \stackrel{\text{def}}{=} A$ Immediate.
Case $P \stackrel{\text{def}}{=} P_1 \implies P_2$
Pick an arbitrary $\iota \in \mathsf{LEnv}$, $l, g, g' \in \mathsf{LState}$ and $\mathcal{I}, \mathcal{I}' \in \mathsf{AMod}$ such that

$$(l, g, \mathcal{I}), \iota \vDash P_1 \lor P_2 \tag{A.187}$$

$$\mathcal{I}' \uparrow^{(g')} (g, \mathcal{I}) \tag{A.188}$$

$$\begin{aligned}&\forall l, g, g' \in \mathsf{LState}.\, \forall \mathcal{I}', \mathcal{I} \in \mathsf{AMod}.\\ &\quad (l, g, \mathcal{I}), \iota \vDash P_1 \land \mathcal{I}' \uparrow^{(g')} (g, \mathcal{I}) \implies (l, g \circ g', \mathcal{I} \cup \mathcal{I}'), \iota \vDash P_1\end{aligned} \tag{I.H1}$$

$$\begin{aligned}&\forall l, g, g' \in \mathsf{LState}.\, \forall \mathcal{I}', \mathcal{I} \in \mathsf{AMod}.\\ &\quad (l, g, \mathcal{I}), \iota \vDash P_2 \land \mathcal{I}' \uparrow^{(g')} (g, \mathcal{I}) \implies (l, g \circ g', \mathcal{I} \cup \mathcal{I}'), \iota \vDash P_2\end{aligned} \tag{I.H2}$$

From (A.187) and the definition of $\vDash$ we know $(l, g, \mathcal{I}), \iota \vDash P_1$ or $(l, g, \mathcal{I}), \iota \vDash P_2$; consequently, from (A.188), (I.H1) and (I.H2) we have: $(l, g \circ g', \mathcal{I} \cup \mathcal{I}'), \iota \vDash P_1$ or $(l, g \circ g', \mathcal{I} \cup \mathcal{I}'), \iota \vDash P_2$. Thus, from the definition of $\vDash$ we have:

$$(l, g \circ g', \mathcal{I} \cup \mathcal{I}'), \iota \vDash P_1 \lor P_2$$

as required.

Cases $P \stackrel{\text{def}}{=} \exists x.\, P'$**;** $P \stackrel{\text{def}}{=} P_1 * P_2$**;** $P \stackrel{\text{def}}{=} P_1 \uplus P_2$**;** $P \stackrel{\text{def}}{=} P_1 \land P_2$
These cases are analogous to the previous case and are omitted here.

Case $P \stackrel{\text{def}}{=} \boxed{P'}_I$
Pick an arbitrary $\iota \in \mathsf{LEnv}$, $s, g \in \mathsf{LState}$ and $\mathcal{I}, \mathcal{I}' \in \mathsf{AMod}$ such that

$$(l, g, \mathcal{I}), \iota \vDash \boxed{P'}_I \tag{A.189}$$

$$\mathcal{I}' \uparrow^{(g')} (g, \mathcal{I}) \tag{A.190}$$
$$\forall l, g, g' \in \mathsf{LState}.\, \forall \mathcal{I}', \mathcal{I} \in \mathsf{AMod}.$$
$$(l, g, \mathcal{I}), \iota \vDash P' \land \mathcal{I}' \uparrow^{(g')} (g, \mathcal{I}) \implies (l, g \circ g', \mathcal{I} \cup \mathcal{I}'), \iota \vDash P' \tag{I.H}$$

From (A.189) and the definition of $\vDash$ we have:

$$l = \mathbf{0}_{\mathsf{L}} \land \exists s', r'.\, g = s' \circ r' \land s', \iota \vDash_{g,\mathcal{I}} P' \land \mathcal{I}\downarrow \left(s', r', \langle\!| I |\!\rangle_\iota\right)$$

Thus from (A.190) and Lemma 23 we have

$$l = \mathbf{0}_{\mathsf{L}} \land \exists s', r'.\, g \circ g' = s' \circ r' \circ g' \land s', \iota \vDash_{g \circ g', \mathcal{I} \cup \mathcal{I}'} P' \land$$
$$\mathcal{I}\downarrow \left(s', r', \langle\!| I |\!\rangle_\iota\right)$$

and consequently from (A.190) and (I.H) we have

$$l = \mathbf{0}_{\mathsf{L}} \land \exists s', r'.\, g \circ g' = s' \circ r' \land s', \iota \vDash_{g \circ g', \mathcal{I} \cup \mathcal{I}'} P' \land$$
$$\mathcal{I} \cup \mathcal{I}'\downarrow \left(s', r', \langle\!| I |\!\rangle_\iota\right)$$

That is,

$$(l, g \circ g', \mathcal{I} \cup \mathcal{I}'), \iota \vDash \boxed{P'}_I \tag{A.191}$$

as required. □

Lemma 23. for all $P \in \mathsf{Assn}$, $\iota \in \mathsf{LEnv}$, $s, g, g' \in \mathsf{LState}$ and $\mathcal{I}, \mathcal{I}' \in \mathsf{AMod}$:

$$s, \iota \vDash_{g,\mathcal{I}} P \land \mathcal{I}' \uparrow^{(g')} (g, \mathcal{I}) \implies s, \iota \vDash_{g \circ g', \mathcal{I} \cup \mathcal{I}'} P$$

Proof. By induction on the structure of assertion P.

Case $P \overset{\text{def}}{=} A$ Immediate.
Case $P \overset{\text{def}}{=} P_1 \implies P_2$
Pick an arbitrary $\iota \in \mathsf{LEnv}$, $s, g, g' \in \mathsf{LState}$ and $\mathcal{I}, \mathcal{I}' \in \mathsf{AMod}$ such that

$$s, \iota \vDash_{g,\mathcal{I}} P_1 \implies P_2 \tag{A.192}$$
$$\mathcal{I}' \uparrow^{(g')} (g, \mathcal{I}) \tag{A.193}$$
$$\forall s, g, g' \in \mathsf{LState}.\, \forall \mathcal{I}', \mathcal{I} \in \mathsf{AMod}.$$
$$s, \iota \vDash_{g,\mathcal{I}} P_1 \land \mathcal{I}' \uparrow^{(g')} (g, \mathcal{I}) \implies s, \iota \vDash_{g \circ g', \mathcal{I} \cup \mathcal{I}'} P_1 \tag{I.H1}$$
$$\forall s, g, g' \in \mathsf{LState}.\, \forall \mathcal{I}', \mathcal{I} \in \mathsf{AMod}.$$
$$s, \iota \vDash_{g,\mathcal{I}} P_2 \land \mathcal{I}' \uparrow^{(g')} (g, \mathcal{I}) \implies s, \iota \vDash_{g \circ g', \mathcal{I} \cup \mathcal{I}'} P_2 \tag{I.H2}$$

From (A.192) and the definition of $\vDash_{g,\mathcal{I}}$ we know $s, \iota \vDash_{g,\mathcal{I}} P_1$ implies $P_2 \vDash_{g,\mathcal{I}}$; consequently, from (A.193), (I.H1) and (I.H2) we have: $s, \iota \vDash_{g \circ g', \mathcal{I} \cup \mathcal{I}'} P_1$ implies $s, \iota \vDash_{g \circ g', \mathcal{I} \cup \mathcal{I}'} P_2$. Thus, from the definition of $\vDash_{g \circ g', \mathcal{I} \cup \mathcal{I}'}$ we have:

$$s, \iota \vDash_{g \circ g', \mathcal{I} \cup \mathcal{I}'} P_1 \implies P_2$$

as required.

Cases $P \overset{\text{def}}{=} \exists x.\, P'$; $P \overset{\text{def}}{=} P_1 * P_2$; $P \overset{\text{def}}{=} P_1 \circledast P_2$
These cases are analogous to the previous case and are omitted here.

Case $P \overset{\text{def}}{=} \boxed{P'}_I$
Pick an arbitrary $\iota \in \mathsf{LEnv}$, $s, g \in \mathsf{LState}$ and $\mathcal{I}, \mathcal{I}' \in \mathsf{AMod}$ such that

$$s, \iota \vDash_{g,\mathcal{I}} \boxed{P'}_I \tag{A.194}$$

$$\mathcal{I}' \uparrow^{(g')} (g, \mathcal{I}) \tag{A.195}$$

$$\begin{aligned} &\forall s, g, g' \in \mathsf{LState}.\, \forall \mathcal{I}', \mathcal{I} \in \mathsf{AMod}. \\ &\quad s, \iota \vDash_{g,\mathcal{I}} P' \wedge \mathcal{I}' \uparrow^{(g')} (g, \mathcal{I}) \implies s, \iota \vDash_{g \circ g', \mathcal{I} \cup \mathcal{I}'} P' \end{aligned} \tag{I.H}$$

From (A.194) and the definition of $\vDash_{g,\mathcal{I}}$ we have:

$$s = \mathbf{0}_{\mathsf{L}} \wedge \exists s', r'.\, g = s' \circ r' \wedge s', \iota \vDash_{g,\mathcal{I}} P' \wedge \mathcal{I} \downarrow \left(s', r', \langle\!| I |\!\rangle_\iota\right)$$

Thus from (A.195) and (I.H) we have

$$\begin{aligned} &s = \mathbf{0}_{\mathsf{L}} \wedge \exists s', r'.\, g \circ g' = s' \circ r' \wedge s', \iota \vDash_{g \circ g', \mathcal{I} \cup \mathcal{I}'} P' \wedge \\ &\mathcal{I} \cup \mathcal{I}' \downarrow \left(s', r', \langle\!| I |\!\rangle_\iota\right) \end{aligned}$$

That is,

$$s, \iota \vDash_{g, \mathcal{I} \cup \mathcal{I}'} \boxed{P'}_I$$

as required. □

Lemma 24. for all $P \in \mathsf{Assn}$, $\iota \in \mathsf{LEnv}$, $w, w' \in \mathsf{World}$,

$$w, \iota \vDash P \wedge (w, w') \in R^{\mathsf{s}} \implies w', \iota \vDash P$$

Proof. From the definition of R^{s}, it then suffices to show that for all $P \in \mathsf{Assn}$, $\iota \in \mathsf{LEnv}$, $l, g, \in \mathsf{LState}$ and $\mathcal{I}, \mathcal{I}' \in \mathsf{AMod}$:

$$(l, g, \mathcal{I}), \iota \vDash P \wedge \mathcal{I}' \uparrow^{(\mathbf{0}_{\mathsf{L}})} (g, \mathcal{I}) \implies (l, g, \mathcal{I} \cup \mathcal{I}'), \iota \vDash P$$

This follows immediately from the more general result established in Lemma 22 from (A.187) to (A.191). □

Lemma 25. for all $P \in \mathsf{Assn}$, $\iota \in \mathsf{LEnv}$, $s, g \in \mathsf{LState}$ and $\mathcal{I}, \mathcal{I}' \in \mathsf{AMod}$:

$$(s, g, \mathcal{I}), \iota \vDash P \wedge \mathcal{I}' \uparrow^{(s)} (g, \mathcal{I}) \implies s, \iota \vDash_{g \circ s, \mathcal{I} \cup \mathcal{I}'} P$$

Proof. By induction on the structure of assertion P.

Case $P \overset{\text{def}}{=} A$ Immediate.
Case $P \overset{\text{def}}{=} P_1 \implies P_2$
Pick an arbitrary $\iota \in \mathsf{LEnv}$, $s, g \in \mathsf{LState}$ and $\mathcal{I}, \mathcal{I}' \in \mathsf{AMod}$ such that

$$(s, g, \mathcal{I}), \iota \vDash P_1 \implies P_2 \tag{A.196}$$

$$\mathcal{I}' \uparrow^{(s)} (g, \mathcal{I}) \tag{A.197}$$

$$\forall s, g \in \mathsf{LState}.\, \forall \mathcal{I}, \mathcal{I}' \in \mathsf{AMod}.$$

$$(s, g, \mathcal{I}), \iota \vDash P_1 \wedge \mathcal{I}' \uparrow^{(s)} (g, \mathcal{I}) \implies s, \iota \vDash_{g \circ s, \mathcal{I} \cup \mathcal{I}'} P_1 \tag{I.H1}$$

$$\forall s, g \in \mathsf{LState}.\, \forall \mathcal{I}, \mathcal{I}' \in \mathsf{AMod}.$$

$$(s, g, \mathcal{I}), \iota \vDash P_2 \wedge \mathcal{I}' \uparrow^{(s)} (g, \mathcal{I}) \implies s, \iota \vDash_{g \circ s, \mathcal{I} \cup \mathcal{I}'} P_2 \tag{I.H2}$$

From (A.196) and the definition of $\vDash$ we know $(s, g, \mathcal{I}), \iota \vDash P_1$ implies $(s, g, \mathcal{I}), \iota \vDash P_2$; consequently, from (A.197), (I.H1) and (I.H2) we have: $s, \iota \vDash_{g \circ s, \mathcal{I} \cup \mathcal{I}'} P_1$ implies $s, \iota \vDash_{g \circ s, \mathcal{I} \cup \mathcal{I}'} P_2$. Thus, from the definition of $\vDash_{g \circ s, \mathcal{I} \cup \mathcal{I}'}$ we have:

$$s, \iota \vDash_{g \circ s, \mathcal{I} \cup \mathcal{I}'} P_1 \implies P_2$$

as required.

Cases $P \overset{\text{def}}{=} \exists x.\, P'$**;** $P \overset{\text{def}}{=} P_1 * P_2$**;** $P \overset{\text{def}}{=} P_1 \circledast P_2$
These cases are analogous to the previous case and are omitted here.

Case $P \overset{\text{def}}{=} \boxed{P'}_I$
Pick an arbitrary $\iota \in \mathsf{LEnv}$, $s, g \in \mathsf{LState}$ and $\mathcal{I}, \mathcal{I}' \in \mathsf{AMod}$ such that

$$(s, g, \mathcal{I}), \iota \vDash \boxed{P'}_I \tag{A.198}$$

$$\mathcal{I}' \uparrow^{(s)} (g, \mathcal{I}) \tag{A.199}$$

From (A.198) and the definition of $\vDash$ we have:

$$s = \mathbf{0}_{\mathsf{L}} \wedge \exists s', r'.\, g = s' \circ r' \wedge s', \iota \vDash_{g, \mathcal{I}} P' \wedge \mathcal{I} \downarrow \left(s', r', \langle\!| I |\!\rangle_\iota\right)$$

Thus from (A.199) we have

$$s = \mathbf{0}_{\mathsf{L}} \wedge \exists s', r'.\, g = s' \circ r' \wedge s', \iota \vDash_{g, \mathcal{I}} P' \wedge \mathcal{I} \cup \mathcal{I}' \downarrow \left(s', r' \circ s, \langle\!| I |\!\rangle_\iota\right)$$

and consequently from (A.199), Lemma 23 and since $s = \mathbf{0}_{\mathsf{L}}$

$$\begin{aligned}&s = \mathbf{0}_{\mathsf{L}} \land \exists s', r'.\, g \circ s = s' \circ r' \land s', \iota \vDash_{g \circ s, \mathcal{I} \cup \mathcal{I}'} P' \\ &\land \mathcal{I} \cup \mathcal{I}' \downarrow \left(s', r' \circ s, \langle\!| I |\!\rangle_{\iota}\right)\end{aligned}$$

That is,

$$s, \iota \vDash_{g \circ s, \mathcal{I} \cup \mathcal{I}'} \boxed{P'}_{I}$$

as required. $\square$

Lemma 26. Given any separation algebra $(\mathbb{A}, \bullet_{\mathbb{A}}, \mathbf{0}_{\mathbb{A}})$ with the cross-split property, for any $a, b \in \mathbb{A}$:

$$\exists c \in \mathbb{A}.\, a \leq b \bullet_{\mathbb{A}} c \iff a \sqcap_{\mathbb{A}} b \neq \emptyset$$

Proof $\implies$. We proceed with proof by contradiction. Take arbitrary $a, b, c \in \mathbb{A}$ such that

$$a \leq b \circ c \tag{A.200}$$

and assume

$$a \sqcap_{\mathbb{A}} b = \emptyset \tag{A.201}$$

From (A.201) and by definition of $\sqcap_{\mathbb{A}}$, we have:

$$\neg\exists d, e, f, g.\, a = d \bullet_{\mathbb{A}} e \wedge b = e \bullet_{\mathbb{A}} f \wedge d \bullet_{\mathbb{A}} e \bullet_{\mathbb{A}} f = g \tag{A.202}$$

From (A.200) we have $\exists h.\, a \bullet_{\mathbb{A}} h = b \bullet_{\mathbb{A}} c$ and consequently by the cross-split property we have:

$$\exists ab, ac, hb, hc, t.\, ab \bullet_{\mathbb{A}} ac = a \tag{A.203}$$

$$hb \bullet_{\mathbb{A}} hc = h \tag{A.204}$$

$$ab \bullet_{\mathbb{A}} hb = b \tag{A.205}$$

$$ac \bullet_{\mathbb{A}} hc = c \tag{A.206}$$

$$t = ab \bullet_{\mathbb{A}} ac \bullet_{\mathbb{A}} hb \bullet_{\mathbb{A}} hc \tag{A.207}$$

From (A.207) we have:

$$\exists s.\, ab \bullet_{\mathbb{A}} ac \bullet_{\mathbb{A}} hb = s \tag{A.208}$$

From (A.203), (A.205) and (A.208) we have:

$$\exists d, e, f, g.\, a = d \bullet_{\mathbb{A}} e \wedge b = e \bullet_{\mathbb{A}} f \wedge d \bullet_{\mathbb{A}} e \bullet_{\mathbb{A}} f = g \tag{A.209}$$

From (A.202) and (A.209) we derive a contradiction and can hence deduce:

$$a \sqcap_{\mathbb{A}} b \neq \emptyset$$

as required.

Proof $\Leftarrow$. Take arbitrary $a, b \in \mathbb{A}$ such that

$$a \sqcap_{\mathbb{A}} b \neq \emptyset$$

Then by definition of $\sqcap_{\mathbb{A}}$ we have:

$$\begin{aligned}\exists d, e, f \in \mathbb{A}.\, a = d \bullet_{\mathbb{A}} e \\ b = e \bullet_{\mathbb{A}} f \\ d \bullet_{\mathbb{A}} e \bullet_{\mathbb{A}} f \text{ is defined}\end{aligned}$$

and thus by definition of $\leq$ we have $a \leq b \bullet_{\mathbb{A}} d$ and consequently

$$\exists c.\, a \leq b \bullet_{\mathbb{A}} c$$

as required. □

Lemma 27. Given any separation algebra $(\mathcal{M}, \bullet_{\mathcal{M}}, \mathbf{0}_{\mathcal{M}})$ with the cross-split property:

$$\forall a, b, c \in \mathcal{M}.\, a \leq b \bullet_{\mathcal{M}} c \implies \exists m, n.\, a = m \bullet_{\mathcal{M}} n \;\wedge\; m \leq b \;\wedge\; n \leq c$$

Proof. Pick an arbitrary $a, b, c \in \mathcal{M}$. Since $a \leq b \bullet_{\mathcal{M}} c$, we know $\exists d \in \mathcal{M}$. such that:

$$a \bullet_{\mathcal{M}} d = b \bullet_{\mathcal{M}} c \tag{A.210}$$

By the cross-split property of $\mathcal{M}$, We can deduce: $\exists ab, ac, db, dc \in \mathcal{M}$. such that:

$$a = ab \bullet_{\mathcal{M}} ac \tag{A.211}$$
$$b = ab \bullet_{\mathcal{M}} db \tag{A.212}$$
$$c = ac \bullet_{\mathcal{M}} dc \tag{A.213}$$
$$d = db \bullet_{\mathcal{M}} dc$$

Since $ab \leq b$ (A.212) and $ac \leq c$ (A.213), from (A.211) we can deduce:

$$\exists m, n \in \mathcal{M}.\, a = m \bullet_{\mathcal{M}} n \;\wedge\; m \leq b \;\wedge\; n \leq c$$

as required. □

Lemma 28. Given any separation algebra $(\mathcal{M}, \bullet_{\mathcal{M}}, \mathbf{0}_{\mathcal{M}})$,

$$\forall a, b, c, d \in \mathcal{M}.\, a \bullet_{\mathcal{M}} b = d \;\wedge\; c \leq b \implies \exists f \in \mathcal{M}.\, a \bullet_{\mathcal{M}} c = f$$

Proof. Pick an arbitrary $a, b, c, d \in \mathcal{M}$ such that:

$$a \bullet_{\mathcal{M}} b = d \tag{A.214}$$
$$c \leq b \tag{A.215}$$

From (A.215), we have:

$$\exists e \in \mathcal{M}.\, c \bullet_{\mathcal{M}} e = b \tag{A.216}$$

and consequently from (A.214) we have:

$$a \bullet_{\mathcal{M}} c \bullet_{\mathcal{M}} e = d \tag{A.217}$$

Since $e \leq d$ (A.217), we have:

$$\exists f \in \mathcal{M}.\, e \bullet_{\mathcal{M}} f = d \tag{A.218}$$

From (A.217), (A.218) and cancellativity of separation algebras we have:

$$a \bullet_{\mathcal{M}} c = f \tag{A.219}$$

and thus

$$\exists f \in \mathcal{M}.\, a \bullet_{\mathcal{M}} c = f \tag{A.220}$$

as required. $\square$