A Null Pointer Dereference Bug in OpenSSL (CVE-2014-0198)

$$
\begin{array}{l}
\texttt{NPD}(b) \triangleq \texttt{local}\ x_b, x_p\ \texttt{in} \\
\qquad x_b := [b]; \\
\qquad \{\texttt{assume}(x_b{=}\texttt{null}); \texttt{setup\_write\_buffer}(b)\} + \{\texttt{assume}(x_b{\neq}\texttt{null}); \texttt{skip}\}; \\
\qquad \texttt{dispatch\_alert}(b) + \texttt{skip}; \\
\qquad x_p := [b];\ \textsc{l}_{rp}\colon [x_p] := 666
\end{array}
$$

Fig. 9. The NPD program from OpenSSL adapted to the ISL language

We consider a null-pointer-dereference bug in OpenSSL, adapted to our ISL language as the $\texttt{NPD}(b)$ program in Fig. 9. The $\texttt{NPD}(b)$ program makes calls to the $\texttt{setup\_write\_buffer}(b)$ and $\texttt{dispatch\_alert}(b)$ procedures, assumed to be inlined within $\texttt{NPD}(b)$, as before. For brevity, we omit the code of these two procedures, and note that while $\texttt{setup\_write\_buffer}(b)$ always ensures that the buffer at b is allocated, $\texttt{dispatch\_alert}(b)$ may accidentally deallocate the buffer at b and set it to $\texttt{null}$, causing a null-pointer-dereference error later. We thus assume the following specifications for these procedures:

$$[b \mapsto e]\ \texttt{setup\_write\_buffer}(b)\ [ok\colon \exists l_2.\ b \mapsto l_2 * l_2 \mapsto e] \qquad \text{(NPD-Setup)}$$

$$[b \mapsto l_2 * l_2 \mapsto -]\ \texttt{dispatch\_alert}(b)\ [ok\colon b \mapsto \texttt{null} * l_2 \not\mapsto\ ] \qquad \text{(NPD-Alert)}$$

We can then prove the following error specifications for $\texttt{NPD}(b)$:

$$[b \mapsto \texttt{null}]\ \texttt{NPD}(b)\ [er(\textsc{l}_{rp})\colon \exists l_2.\ b \mapsto \texttt{null} * l_2 \not\mapsto\ ] \qquad \text{(NPD-Er-1)}$$

$$[b \mapsto l_2 * l_2 \mapsto -]\ \texttt{NPD}(b)\ [er(\textsc{l}_{rp})\colon b \mapsto \texttt{null} * l_2 \not\mapsto\ ] \qquad \text{(NPD-Er-2)}$$

(NPD-Er-1) describes the case where the buffer at b is originally unallocated and is subsequently allocated by $\texttt{setup\_write\_buffer}(b)$, only to be deallocated by $\texttt{dispatch\_alert}(b)$ shortly after, causing a null-pointer-dereference error at $\textsc{l}_{rp}$. Analogously, (NPD-Er-2) describes the case where the buffer is initially allocated and later deallocated by $\texttt{dispatch\_alert}(b)$, causing an error at $\textsc{l}_{rp}$.

The proofs of (NPD-Er-1) and (NPD-Er-2) are straightforward. A proof sketch of (NPD-Er-1) is given in Fig. 10; the (NPD-Er-2) proof is analogous and omitted.

$[b \mapsto \texttt{null}]$

$\quad \texttt{local}\ x_b, x_p\ \texttt{in}$

$\quad\quad x_b := [b];$ // (LOAD)

$\quad\quad [ok : x_b = \texttt{null} * b \mapsto \texttt{null}]$

$\quad\quad \{\ \ \texttt{assume}(x_b{=}\texttt{null});$ // (ASSUME)

$\quad\quad\quad [ok : x_b{=}\texttt{null} * b \mapsto \texttt{null}]$

$\quad\quad\quad \texttt{setup\_write\_buffer}(b);$ // (NPD-SETUP)

$\quad\quad\quad [ok : x_b{=}\texttt{null} * \exists l_2. b \mapsto l_2 * l_2 \mapsto -]$

$\quad\quad \} + \{\ldots\};$ // (CHOICE)

$\quad\quad [ok : x_b{=}\texttt{null} * \exists l_2.\ b \mapsto l_2 * l_2 \mapsto -]$

$\quad\quad\quad [ok : b \mapsto l_2 * l_2 \mapsto -]$

$\quad\quad\quad (\texttt{dispatch\_alert}(b);$ // (NPD-ALERT)

$\quad\quad\quad [ok : b \mapsto \texttt{null} * l_2 \not\mapsto\ ]$

$\quad\quad [ok : x_b{=}\texttt{null} * \exists l_2.\ b \mapsto \texttt{null} * l_2 \not\mapsto\ ]$ // (EXIST, 4.2)

$\quad\quad + \ldots);$ // (CHOICE)

$\quad\quad [ok : x_b{=}\texttt{null} * \exists l_2.\ b \mapsto \texttt{null} * l_2 \not\mapsto\ ]$

$\quad\quad x_p := [b];$ // (LOAD)

$\quad\quad [ok : x_b{=}x_p{=}\texttt{null} * \exists l_2.\ b \mapsto \texttt{null} * l_2 \not\mapsto\ ]$

$\quad\quad \mathrm{L}_{rp}\colon [x_p] := 666$ // (STORENULL)

$\quad\quad [er(\mathrm{L}_{rp}) : x_b{=}x_p{=}\texttt{null} * \exists l_2.\ b \mapsto \texttt{null} * l_2 \not\mapsto\ ]$

$\quad\quad$ // (LOCAL, CONS)

$[er(\mathrm{L}_{rp}) : \exists l_2.\ b \mapsto \texttt{null} * l_2 \not\mapsto\ ]$

Fig. 10. A proof sketch of NPD-ER-1

B Soundness

Definition 1.

$$s_1 \sim_A s_2 \stackrel{def}{\iff} \forall x \in A.\ s_1(x){=}s_2(x)$$

Proposition 1. *For all assertions p and all s, s', h, if $(s, h) \in p$ and $s \sim_{\mathsf{fv}(p)} s'$, then $(s', h) \in p$.*
For all ϵ, $\mathbb{C}$, x, v, (s_1, h_1) and (s_2, h_2), if $((s_1, h_1), (s_2, h_2)) \in [\![\mathbb{C}]\!]$ and $x \notin \mathsf{fv}(\mathbb{C})$, then $((s_1[x \mapsto v], h_1), (s_2[x \mapsto v], h_2)) \in [\![\mathbb{C}]\!]$.

Lemma 1. *For all $p, \mathbb{C}, q, \epsilon$, if $\vdash [p]\ \mathbb{C}\ [\epsilon : q]$ holds, then:*

$$\begin{aligned}&\forall (s_q, h_q) \in q.\ \forall h.\ h_q \# h \implies\\ &\quad \exists (s_p, h_p) \in p.\ s_p \sim_{\overline{\mathsf{mod}(\mathbb{C})}} s_q \wedge \big((s_p, h_p \uplus h), (s_q, h_q \uplus h)\big) \in [\![\mathbb{C}]\!]\epsilon\end{aligned}$$

where $h_q \# h \stackrel{def}{\iff} dom(h_q) \cap dom(h){=}\emptyset$ denotes that h_q and h are compatible *in that their composition is defined.*

Proof. We proceed by induction on the structure of incorrectness triples. In what follows we write h_0 to denote an empty heap (i.e., $dom(h_0) = \emptyset$).

Case Skip
Pick an arbitrary $\sigma_q{=}(s, h_q) \in \mathsf{emp}$ and an arbitrary h such that $h_q \# h$. It then suffices to show that $((s, h_q \uplus h), (s, h_q \uplus h)) \in [\![\texttt{skip}]\!]ok$, which follows from the semantics of `skip` immediately.

Case Assign
Pick an arbitrary x, e, h and $(s_q, h_q) \in q$ such that $h_q \uplus h$. We then know $h_q{=}h_0$ and $s_q(x){=}s_q(e[x'/x])$. Let $s_p{=}s_q[x \mapsto s_q(x')]$. By definition we then know that $s_p \sim_{\overline{\mathsf{mod}(\mathbb{C})}} s_q$ and $(s_p, h_q) \in p$. It then suffices to show that $((s_p, h_q \uplus h), (s_q, h_q \uplus h)) \in [\![x := e]\!]ok$, which follows from the semantics of $x := e$ immediately.

The proof of Havoc is analogous and omitted here.

Case Load
Pick an arbitrary x, y and $(s_q, h_q) \in q$. Pick an arbitrary h such that $h_q \# h$. We then know there exist l, v such that $s_q(x){=}s_q(e[x'/x]){=}v$, $s_q(y){=}l$ and $h_q \triangleq [l \mapsto v]$. Let $s_p \triangleq s_q[x \mapsto s_q(x')]$. By definition we then know that $s_p \sim_{\overline{\mathsf{mod}(\mathbb{C})}} s_q$ and $(s_p, h_q) \in p$. It then suffices to show $((s_p, h_q \uplus h), (s_q, h_q \uplus h)) \in [\![x := [y]]\!]ok$, which follows from the semantics of $x := [y]$ immediately.

Case LoadEr
Pick an arbitrary x, y and $(s_q, h_q) \in q$. Pick an arbitrary h such that $h_q \# h$. We then know there exist l such that $s_q(y){=}l$ and $h_q \triangleq [l \mapsto \bot]$. By definition we then know that $s_q \sim_{\overline{\mathsf{mod}(\mathbb{C})}} s_q$ and $(s_q, h_q) \in p$. It then suffices to show $((s_q, h_q \uplus h), (s_q, h_q \uplus h)) \in [\![x := [y]]\!]er(\textsc{l})$, which follows from the semantics of $x := [y]$ immediately.

Case LoadNull
Pick an arbitrary x, y and $(s_q, h_q) \in q$. Pick an arbitrary h such that $h_q \# h$. We then know $h_q = h_0$ and $s_q(y) = \texttt{null}$. By definition we then know that $s_q \sim_{\overline{\mathsf{mod}(\mathbb{C})}} s_q$ and $(s_q, h_q) \in p$. It then suffices to show $((s_q, h_q \uplus h), (s_q, h_q \uplus h)) \in [\![x := [y]]\!] er(\textsc{l})$, which follows from the semantics of $x := [y]$ immediately.

The proofs of the Store, StoreEr and StoreNull cases are analogous to those of Load, LoadEr and LoadNull respectively, and are omitted here.

Case Alloc1
Pick an arbitrary x and $(s_q, h_q) \in q$. We then know there exists l and $v \in \textsc{Val}$ such that $s_q(x) = l$ and $h_q \triangleq [l \mapsto v]$. Pick an arbitrary h such that $h_q \# h$. Let $s_p \triangleq s_q[x \mapsto s_q(x')]$, $h_p = h_0$. By definition we then know that $s_p \sim_{\overline{\mathsf{mod}(\mathbb{C})}} s_q$ and $(s_p, h_p) \in p$. Since $h_q \# h$ and $dom(h_p) \subseteq dom(h_q)$, from the definition of $\uplus$ we also know that $h_p \# h$. It then suffices to show that $((s_p, h_p \uplus h), (s_q, h_q \uplus h)) \in [\![x := \texttt{alloc()}]\!] ok$, which follows from the semantics of $x := \texttt{alloc()}$.

Case Alloc2
Pick an arbitrary x, y and $(s_q, h_q) \in q$. We then know there exists l and $v \in \textsc{Val}$ such that $s_q(x) = s_q(y) = l$ and $h_q \triangleq [l \mapsto v]$. Pick an arbitrary h such that $h_q \# h$. Let $s_p \triangleq s_q[x \mapsto s_q(x')]$, $h_p = [l \mapsto \bot]$. By definition we then know that $s_p \sim_{\overline{\mathsf{mod}(\mathbb{C})}} s_q$ and $(s_p, h_p) \in p$. Since $h_q \# h$ and $dom(h_p) = dom(h_q)$, from the definition of $\uplus$ we also know that $h_p \# h$. It then suffices to show that $((s_p, h_p \uplus h), (s_q, h_q \uplus h)) \in [\![x := \texttt{alloc()}]\!] ok$, which follows from the semantics of $x := \texttt{alloc()}$.

Case Free
Pick an arbitrary x and $(s_q, h_q) \in q$. We then know there exists l such that $s_q(x) = l$ and $h_q \triangleq [l \mapsto \bot]$. Pick an arbitrary h such that $h_q \# h$. Let $h_p = [l \mapsto s_q(e)]$. By definition we then know that $s_q \sim_{\overline{\mathsf{mod}(\mathbb{C})}} s_q$ and $(s_q, h_p) \in p$. Since $h_q \# h$ and $dom(h_p) = dom(h_q)$, from the definition of $\uplus$ we also know that $h_p \# h$. It then suffices to show that $((s_q, h_p \uplus h), (s_q, h_q \uplus h)) \in [\![\texttt{free}(x)]\!] ok$, which follows from the semantics of $\texttt{free}(x)$ immediately.

Case FreeEr
Pick an arbitrary x and $(s_q, h_q) \in q$. We then know there exists l such that $s_q(x) = l$ and $h_q \triangleq [l \mapsto \bot]$. Pick an arbitrary h such that $h_q \# h$. By definition we then know that $s_q \sim_{\overline{\mathsf{mod}(\mathbb{C})}} s_q$. It then suffices to show that $((s_q, h_q \uplus h), (s_q, h_q \uplus h)) \in [\![\texttt{free}(x)]\!] er(\textsc{l})$, which follows from the semantics of $\texttt{free}(x)$ immediately.

Case FreeNull
Pick an arbitrary x and $(s_q, h_q) \in q$. We then know $h_q = h_0$ and $s_q(x) = \texttt{null}$. Pick an arbitrary h such that $h_q \# h$. By definition we then know that $s_q \sim_{\overline{\mathsf{mod}(\mathbb{C})}} s_q$. It then suffices to show that $((s_q, h_q \uplus h), (s_q, h_q \uplus h)) \in [\![\texttt{free}(x)]\!] er(\textsc{l})$, which

follows from the semantics of $\texttt{free}(x)$ immediately.

Case Error
Pick an arbitrary $(s_q, h_q) \in q$. We then know that $h_q \triangleq h_0$. Pick an arbitrary h such that $h_q \# h$. As $(s_q, h_q) \in q$, it then suffices to show that $((s_q, h_q \uplus h), (s_q, h_q \uplus h)) \in [\![\texttt{error}]\!]er(\textsc{l})$, which follows from the semantics of $\texttt{error}$ immediately.

Case Assume
Pick an arbitrary $(s_q, h_q) \in q$. We then know that $s_q(B) \neq 0$. Pick an arbitrary h such that $h_q \# h$. Since $q{=}p \wedge B$ and $(s_q, h_q) \in q$, we also have $(s_q, h_q) \in p$. By definition we know that $s_q \sim_{\overline{\mathsf{mod}(\mathbb{C})}} s_q$. It thus suffices to show that $((s_q, h_q \uplus h), (s_q, h_q \uplus h)) \in [\![\texttt{assume}(B)]\!]ok$, which follows from the semantics of $\texttt{assume}(B)$ immediately.

Case Local
Pick an arbitrary x, h and $(s_q, h_q) \in \exists x.\, q$ such that $h_q \# h$. From the semantics of assertions we then know that there exists v and s'_q such that $s'_q = s_q[x \mapsto v]$ and $(s'_q, h_q) \in q$. Since from the premise of Local we have $[p]\ \mathbb{C}\ [\epsilon : q]$, from the inductive hypothesis we know there exist s'_p, h_p such that $s'_p \sim_{\overline{\mathsf{mod}(\mathbb{C})}} s'_q$, $(s'_p, h_p) \in p$ and $((s'_p, h_p \uplus h), (s'_q, h_q \uplus h)) \in [\![\mathbb{C}]\!]\epsilon$. Let $s_p = s'_p[x \mapsto s_q(x)]$. Note that since $s_p[x \mapsto s'_p(x)] = s'_p$ and $(s'_p, h_p) \in p$, from the semantics of assertions we have $(s_p, h_p) \in \exists x.\, p$. On the other hand, since $s_p(x) = s_q(x)$ and $((s'_p, h_p \uplus h), (s'_q, h_q \uplus h)) \in [\![\mathbb{C}]\!]\epsilon$, from the definitions of $[\![.]\!]$, s_p, s'_p, s_q and s'_q we also have $((s_p, h_p \uplus h), (s_q, h_q \uplus h)) \in [\![\texttt{local } x \texttt{ in } \mathbb{C}]\!]\epsilon$. Moreover, since $s'_p \sim_{\overline{\mathsf{mod}(\mathbb{C})}} s'_q$ and $s_p = s'_p[x \mapsto s_q(x)]$ and thus $s_q(x) = s_p(x)$, we also have $s_p \sim_{\overline{\mathsf{mod}(\texttt{local } x \texttt{ in } \mathbb{C})}} s_q$, as required.

Case Exist
Pick an arbitrary x, h and $(s_q, h_q) \in \exists x.\, q$ such that $h_q \# h$. From the semantics of assertions we then know that there exists v and s'_q such that $s'_q = s_q[x \mapsto v]$ and $(s'_q, h_q) \in q$. Since from the premise of Exist we have $[p]\ \mathbb{C}\ [\epsilon : q]$, from the inductive hypothesis we know there exist s'_p, h_p such that $s'_p \sim_{\overline{\mathsf{mod}(\mathbb{C})}} s'_q$, $(s'_p, h_p) \in p$ and $((s'_p, h_p \uplus h), (s'_q, h_q \uplus h)) \in [\![\mathbb{C}]\!]\epsilon$. Let $s_p = s'_p[x \mapsto s_q(x)]$. Note that since $s_p[x \mapsto s'_p(x)] = s'_p$ and $(s'_p, h_p) \in p$, from the semantics of assertions we have $(s_p, h_p) \in \exists x.\, p$. On the other hand, since $s_p(x) = s_q(x)$ and $((s'_p, h_p \uplus h), (s'_q, h_q \uplus h)) \in [\![\mathbb{C}]\!]\epsilon$, and since $x \notin \mathsf{fv}(\mathbb{C})$, from Proposition 1 and the definitions of s_p, s'_p, s_q and s'_q we also have $((s_p, h_p \uplus h), (s_q, h_q \uplus h)) \in [\![\mathbb{C}]\!]\epsilon$. Moreover, since $s'_p \sim_{\overline{\mathsf{mod}(\mathbb{C})}} s'_q$ and $s_p = s'_p[x \mapsto s_q(x)]$ and thus $s_q(x) = s_p(x)$, from the definitions of s_p, s'_p, s_q and s'_q we also have $s_p \sim_{\overline{\mathsf{mod}(\mathbb{C})}} s_q$, as required.

Case Seq1
Pick an arbitrary $(s_q, h_q) \in q$ and h such that $h_q \# h$. Since from the premise of Seq1 we have $[p]\ \mathbb{C}\ [\epsilon : q]$ with $\epsilon \neq ok$, from the inductive hypothesis we know there exist s_p, h_p such that $s_p \sim_{\overline{\mathsf{mod}(\mathbb{C}_1)}} s_q$, $(s_p, h_p) \in p$ and $((s_p, h_p \uplus h), (s_q, h_q \uplus h)) \in$

$[\![\mathbb{C}_1]\!]\epsilon$. As such, since $\mathsf{mod}(\mathbb{C}_1) \subseteq \mathsf{mod}(\mathbb{C}_1;\mathbb{C}_2)$, we know $s_p \sim_{\overline{\mathsf{mod}(\mathbb{C}_1;\mathbb{C}_2)}} s_q$, $(s_p, h_p) \in p$ and from the semantics of $\mathbb{C}_1;\mathbb{C}_2$ we have $((s_p, h_p \uplus h), (s_q, h_q \uplus h)) \in [\![\mathbb{C}_1;\mathbb{C}_2]\!]\epsilon$, as required.

Case Seq2
Pick arbitrary $(s_q, h_q) \in q$ and h such that $h_q \# h$. As from the second premise of Seq2 we have $[r]\ \mathbb{C}_2\ [\epsilon : q]$ and $\mathsf{mod}(\mathbb{C}_2) \subseteq \mathsf{mod}(\mathbb{C}_1;\mathbb{C}_2)$, from the inductive hypothesis we know there exist s_r, h_r such that $s_r \sim_{\overline{\mathsf{mod}(\mathbb{C}_1;\mathbb{C}_2)}} s_q$, $(s_r, h_r) \in r$ and $((s_r, h_r \uplus h), (s_q, h_q \uplus h)) \in [\![\mathbb{C}_2]\!]\epsilon$. Moreover, as from the first premise we have $[p]\ \mathbb{C}_1\ [ok : r]$ and $\mathsf{mod}(\mathbb{C}_1) \subseteq \mathsf{mod}(\mathbb{C}_1;\mathbb{C}_2)$, from the inductive hypothesis we know there exist s_p, h_p such that $s_p \sim_{\overline{\mathsf{mod}(\mathbb{C}_1;\mathbb{C}_2)}} s_r$, $(s_p, h_p) \in p$ and $((s_p, h_p \uplus h), (s_r, h_r \uplus h)) \in [\![\mathbb{C}_1]\!]ok$. As such, since $s_p \sim_{\overline{\mathsf{mod}(\mathbb{C}_1;\mathbb{C}_2)}} s_r$, $s_r \sim_{\overline{\mathsf{mod}(\mathbb{C}_1;\mathbb{C}_2)}} s_q$ we know $s_p \sim_{\overline{\mathsf{mod}(\mathbb{C}_1;\mathbb{C}_2)}} s_q$, $(s_p, h_p) \in p$ and from the semantics of $\mathbb{C}_1;\mathbb{C}_2$ we have $((s_p, h_p \uplus h), (s_q, h_q \uplus h)) \in [\![\mathbb{C}_1;\mathbb{C}_2]\!]\epsilon$, as required.

Case Choice
Pick arbitrary $(s_q, h_q) \in q$ and h such that $h_q \# h$. From the premise of Choice we know there exists $i \in \{1, 2\}$ such that $[p]\ \mathbb{C}_i\ [\epsilon : q]$. As such, from the inductive hypothesis we know there exist s_p, h_p such that $s_p \sim_{\overline{\mathsf{mod}(\mathbb{C}_i)}} s_q$, $(s_p, h_p) \in p$ and $((s_p, h_p \uplus h), (s_q, h_q \uplus h)) \in [\![\mathbb{C}_i]\!]\epsilon$. As such, since $\mathsf{mod}(\mathbb{C}_i) \subseteq \mathsf{mod}(\mathbb{C}_1 + \mathbb{C}_2)$, we know $s_p \sim_{\overline{\mathsf{mod}(\mathbb{C}_1+\mathbb{C}_2)}} s_q$, $(s_p, h_p) \in p$ and from the semantics of $\mathbb{C}_1 + \mathbb{C}_2$ we have $((s_p, h_p \uplus h), (s_q, h_q \uplus h)) \in [\![\mathbb{C}_1 + \mathbb{C}_2]\!]\epsilon$, as required.

Case Loop1
Pick an arbitrary $(s_q, h_q) \in q$ and an arbitrary h such that $h_q \# h$. It then suffices to show that $((s_q, h_q \uplus h), (s_q, h_q \uplus h)) \in [\![\mathbb{C}^\star]\!]ok$, which follows from the semantics of $\mathbb{C}^\star$ immediately.

Case Loop2
Pick arbitrary $(s_q, h_q) \in q$ and h such that $h_q \# h$. From the premise of Loop2 we have $[p]\ \mathbb{C}^\star;\mathbb{C}\ [\epsilon : q]$ and thus from the inductive hypothesis we know there exists s_p, h_p such that $s_p \sim_{\overline{\mathsf{mod}(\mathbb{C};\mathbb{C}^\star)}} s_q$, $(s_p, h_p) \in p$ and $((s_p, h_p \uplus h), (s_q, h_q \uplus h)) \in [\![\mathbb{C}^\star;\mathbb{C}]\!]\epsilon$. Moreover, by definition we have $\mathsf{mod}(\mathbb{C}^\star;\mathbb{C}) = \mathsf{mod}(\mathbb{C}^\star)$. On the other hand, it is straightforward to show that $[\![\mathbb{C}^\star;\mathbb{C}]\!] = \bigcup_{i \in \mathbb{N}^+} [\![\mathbb{C}^i]\!]$ and thus $[\![\mathbb{C}^\star;\mathbb{C}]\!] \subseteq [\![\mathbb{C}^\star]\!]$. Consequently, we know there exists s_p, h_p such that $s_p \sim_{\overline{\mathsf{mod}(\mathbb{C}^\star)}} s_q$, $(s_p, h_p) \in p$ and $((s_p, h_p \uplus h), (s_q, h_q \uplus h)) \in [\![\mathbb{C}^\star]\!]\epsilon$, as required.

Case Cons
Pick arbitrary $(s_q, h_q) \in q$ and h such that $h_q \# h$. As form the premise of Cons we have $q \Rightarrow q'$, we also know that $(s_q, h_q) \in q'$. On the other hand, from the premise of Cons we have $[p']\ \mathbb{C}\ [\epsilon : q']$ and thus from the inductive hypothesis we know there exist s_p, h_p such that $s_p \sim_{\overline{\mathsf{mod}(\mathbb{C})}} s_q$, $(s_p, h_p) \in p'$ and $((s_p, h_p \uplus h), (s_q, h_q \uplus h)) \in [\![\mathbb{C}]\!]\epsilon$. Moreover, as $p' \Rightarrow p$ and $(s_p, h_p) \in p'$ we also

have $(s_p, h_p) \in p$. That is, we know there exists s_p, h_p such that $s_p \sim_{\overline{\mathsf{mod}(\mathbb{C})}} s_q$, $(s_p, h_p) \in p$ and $((s_p, h_p \uplus h), (s_q, h_q \uplus h)) \in [\![\mathbb{C}]\!]\epsilon$, as required.

Case 4.2

Pick arbitrary $(s_2, h_2) \in q * r$ and h such that $h_2 \# h$. From the definition of $*$ we then know there exists h_q, h_r such that $(s_2, h_q) \in q$, $(s_2, h_r) \in r$ and $h_2 \triangleq h_q \uplus h_r$. From the definition of $\#$ and $\uplus$ we then also have $h_q \# h_r \uplus h$. On the other hand, from the premise of 4.2 we have $[p]\ \mathbb{C}\ [\epsilon : q]$ and thus from the inductive hypothesis we know there exists s_1, h_p such that $s_1 \sim_{\overline{\mathsf{mod}(\mathbb{C})}} s_2$, $(s_1, h_p) \in p$ and $((s_1, h_p \uplus h_r \uplus h), (s_q, h_q \uplus h_r \uplus h)) \in [\![\mathbb{C}]\!]\epsilon$. Moreover, since $s_1 \sim_{\overline{\mathsf{mod}(\mathbb{C})}} s_2$ and as from the premise of 4.2 we have $\mathsf{mod}(\mathbb{C}) \cap \mathsf{fv}(r) = \emptyset$, we also have $s_1 \sim_{\mathsf{fv}(r)} s_2$. Consequently, since $(s_2, h_r) \in r$, from Proposition 1 we have $(s_1, h_r) \in r$. As such from the definition of $*$ we have $(s_1, h_p \uplus h_r) \in p * r$. That is, we know there exists s_1 and $h_1 = h_p \uplus h_r$ such that $s_1 \sim_{\overline{\mathsf{mod}(\mathbb{C})}} s_2$, $(s_1, h_1) \in p * r$ and $((s_1, h_1 \uplus h), (s_q, h_2 \uplus h)) \in [\![\mathbb{C}]\!]\epsilon$, as required.

Case Disj

Pick arbitrary $(s_q, h_q) \in q_1 \vee q_2$ and h such that $h_q \# h$. We then know there exists $i \in \{1, 2\}$ such that $(s_q, h_q) \in q_i$. From the premise of Disj we have $[p_i]\ \mathbb{C}\ [\epsilon : q_i]$ and thus from the inductive hypothesis we know there exists s_p, h_p such that $s_p \sim_{\overline{\mathsf{mod}(\mathbb{C})}} s_q$, $(s_p, h_p) \in p_i$ and $((s_p, h_p \uplus h), (s_q, h_q \uplus h)) \in [\![\mathbb{C}]\!]\epsilon$. Moreover, since $p_i \subseteq p_1 \vee p_2$ and $(s_p, h_p) \in p_i$, we also have $(s_p, h_p) \in p_1 \vee p_2$. That is, we know there exists s_p, h_p such that $s_p \sim_{\overline{\mathsf{mod}(\mathbb{C})}} s_q$, $(s_p, h_p) \in p_1 \vee p_2$ and $((s_p, h_p \uplus h), (s_q, h_q \uplus h)) \in [\![\mathbb{C}]\!]\epsilon$, as required.

Theorem 4 (Soundness). *For all $p, \mathbb{C}, q, \epsilon$, if $\vdash [p]\ \mathbb{C}\ [\epsilon : q]$ holds, then $\models [p]\ \mathbb{C}\ [\epsilon : q]$ also holds.*

Proof. Pick arbitrary $p, \mathbb{C}, q, \epsilon$ such that $\vdash [p]\ \mathbb{C}\ [\epsilon : q]$ holds. Pick an arbitrary $(s_q, h_q) \in q$. It then suffices to show there exists $(s_p, h_p) \in p$ such that $((s_p, h_p), (s_q, h_q)) \in [\![\mathbb{C}]\!]\epsilon$.

From the definition of $\uplus$ and $\#$ we then know that $h_q \# h_0$. As such, from Lemma 1 we know there exists $(s_p, h_p) \in p$ such that $((s_p, h_p \uplus h_0), (s_q, h_q \uplus h_0)) \in [\![\mathbb{C}]\!]\epsilon$. That is, there exists $(s_p, h_p) \in p$ such that $((s_p, h_p), (s_q, h_q)) \in [\![\mathbb{C}]\!]\epsilon$, as required.

$$\texttt{foot}\,(.) : \textsc{Comm} \to \textsc{Exit} \to \mathcal{P}(\textsc{State} \times \textsc{State})$$

$$\begin{aligned}
\texttt{foot}\,(\texttt{skip})\,\epsilon &\triangleq \{((s,h_0),(s,h_0)) \mid s \in \textsc{Store}\}\\
\texttt{foot}\,(x := e)\,ok &\triangleq \{((s,h_0),(s[x \mapsto s(e)],h_0)) \mid s \in \textsc{Store}\}\\
\texttt{foot}\,(x := e)\,er(-) &\triangleq \emptyset\\
\texttt{foot}\,(x := \texttt{*})\,ok &\triangleq \{((s,h_0),(s[x \mapsto v],h_0)) \mid v \in \textsc{Val}\}\\
\texttt{foot}\,(x := \texttt{*})\,er(-) &\triangleq \emptyset\\
\texttt{foot}\,(\texttt{assume(}B\texttt{)})\,ok &\triangleq \{(\sigma,\sigma) \mid \sigma{=}(s,h_0) \land s(B){\neq}0\}\\
\texttt{foot}\,(\texttt{assume(}B\texttt{)})\,er(-) &\triangleq \emptyset\\
\texttt{foot}\,(\texttt{local}\ x\ \texttt{in}\ \mathbb{C})\,\epsilon &\triangleq \left\{((s[x \mapsto v],h),(s'[x \mapsto v],h')) \,\middle|\, \begin{array}{l}((s,h),(s',h')) \in \texttt{foot}\,(\mathbb{C})\,\epsilon\\ \land v \in \textsc{Val}\end{array}\right\}\\
\texttt{foot}\,(\textsc{l}\colon \texttt{error})\,ok &\triangleq \emptyset\\
\texttt{foot}\,(\textsc{l}\colon \texttt{error})\,er(\textsc{l}') &\triangleq \{((\sigma,\sigma) \mid \sigma{=}(s,h_0) \land \textsc{l}{=}\textsc{l}'\}\\
\texttt{foot}\,(\mathbb{C}_1;\mathbb{C}_2)\,\epsilon &\triangleq \{(\sigma,\sigma') \mid \epsilon \neq ok \land (\sigma,\sigma') \in \texttt{foot}\,(\mathbb{C}_1)\,\epsilon\}\\
&\quad \cup \left\{(\sigma_1 \bullet \sigma, \sigma_2 \bullet \sigma') \,\middle|\, \begin{array}{l}\exists \sigma_c.\ (\sigma_1, \sigma' \bullet \sigma_c) \in \texttt{foot}\,(\mathbb{C}_1)\,ok\\ \quad \land\ (\sigma_c \bullet \sigma, \sigma_2) \in \texttt{foot}\,(\mathbb{C}_2)\,\epsilon\end{array}\right\}\\
\texttt{foot}\,(\mathbb{C}_1 + \mathbb{C}_2)\,\epsilon &\triangleq \texttt{foot}\,(\mathbb{C}_1)\,\epsilon \cup \texttt{foot}\,(\mathbb{C}_2)\,\epsilon\\
\texttt{foot}\,(\mathbb{C}^\star)\,\epsilon &\triangleq \{((s,h_0),(s,h_0)) \mid \epsilon{=}ok\} \cup \bigcup_{i \in \mathbb{N}^+} \texttt{foot}\left(\mathbb{C}^i\right)\epsilon\\
\texttt{foot}\,(x := \texttt{alloc()})\,ok &\triangleq \{((s,h),(s[x \mapsto l],[l \mapsto v])) \mid v \in \textsc{Val} \land (h{=}h_0 \lor h{=}[l \mapsto \bot]\}\\
\texttt{foot}\,(x := \texttt{alloc()})\,er(-) &\triangleq \emptyset\\
\texttt{foot}\,(\textsc{l}\colon \texttt{free(}x\texttt{)})\,ok &\triangleq \{((s,[l \mapsto v]),(s,[l \mapsto \bot])) \mid s(x){=}l \land v \in \textsc{Val}\}\\
\texttt{foot}\,(\textsc{l}\colon \texttt{free(}x\texttt{)})\,er(\textsc{l}') &\triangleq \{((s,[l \mapsto \bot]),(s,[l \mapsto \bot])) \mid \textsc{l}{=}\textsc{l}' \land s(x){=}l\}\\
&\quad \cup \{((s,h_0),(s,h_0)) \mid \textsc{l}{=}\textsc{l}' \land s(x){=}\texttt{null}\}\\
\texttt{foot}\,(\textsc{l}\colon x := [y])\,ok &\triangleq \{((s,[l \mapsto v]),(s[x \mapsto v],[l \mapsto v])) \mid s(y){=}l\}\\
\texttt{foot}\,(\textsc{l}\colon x := [y])\,er(\textsc{l}') &\triangleq \{((s,[l \mapsto \bot]),(s,[l \mapsto \bot])) \mid \textsc{l}{=}\textsc{l}' \land s(y){=}l\}\\
&\quad \cup \{((s,h_0),(s,h_0)) \mid \textsc{l}{=}\textsc{l}' \land s(y){=}\texttt{null}\}\\
\texttt{foot}\,(\textsc{l}\colon [x] := y)\,ok &\triangleq \{((s,[l \mapsto v]),(s,[l \mapsto s(y)])) \mid s(x){=}l \land v \in \textsc{Val}\}\\
\texttt{foot}\,(\textsc{l}\colon [x] := y)\,er(\textsc{l}') &\triangleq \{((s,[l \mapsto \bot]),(s,[l \mapsto \bot])) \mid \textsc{l}{=}\textsc{l}' \land s(x){=}l\}\\
&\quad \cup \{((s,h_0),(s,h_0)) \mid \textsc{l}{=}\textsc{l}' \land s(x){=}\texttt{null}\}
\end{aligned}$$

Fig. 11. The local ISL footprints where h_0 denotes an empty heap ($dom(h_0){=}\emptyset$)

C Footprints

ISL Footprints The definition of ISL footprints is given in Fig. 11. Note that the definitions of $\texttt{foot}_{\texttt{SL}}(\mathbb{C})$ and $\texttt{foot}\,(\mathbb{C})\ ok$ agree for all $\mathbb{C}$ with the exception of $\mathbb{C}{=}x := \texttt{alloc()}$ and $\mathbb{C}{=}\texttt{free(}x\texttt{)}$. In the case of $\mathbb{C}{=}x := \texttt{alloc()}$ this is because $\texttt{foot}\,(\mathbb{C})$ additionally allows allocation from a singleton heap with a deallocated location. In the case of $\mathbb{C}$=`free(`x`)` this is because $\texttt{foot}\,(\mathbb{C})$ simply mutates the location at x to record $\perp$ and does not remove it from the heap.

It is straightforward to demonstrate that the footprint of a program is included in its semantics, as captured by the following lemma.

Lemma 2. *For all* $\mathbb{C} \in \textsc{Comm}$ *and* $\epsilon \in \textsc{Exit}$*:* $\texttt{foot}\,(\mathbb{C})\,\epsilon \subseteq [\![\mathbb{C}]\!]\epsilon$.

Proof. Follows by straightforward induction on the structure of $\mathbb{C}$.

We next proceed with several auxiliary lemmas and then prove our footprint theorem (Theorem 5).

Lemma 3 (Cross-split property). *For all* $h_1, h_2, h_3, h_4 \in \textsc{Heap}$*:*

$$h_1 \uplus h_2 = h_3 \uplus h_4 \Rightarrow \exists h_{13}, h_{14}, h_{23}, h_{24}.\ h_1{=}h_{13} \uplus h_{14} \wedge h_2{=}h_{23} \uplus h_{24} \\ \wedge\ h_3{=}h_{13} \uplus h_{23} \wedge h_4{=}h_{14} \uplus h_{24}$$

Proof. Follows from the definition of $\uplus$ *on heaps.*

Lemma 4 (Heap Monotonicity). *For all* $s_1, s_2, h, h_1, h_2, \mathbb{C}, \epsilon$*, if* $((s_1, h_1), (s_2, h_2)) \in [\![\mathbb{C}]\!]\epsilon$ *and* $h_2 \# h$*, then* $((s_1, h_1 \uplus h), (s_2, h_2 \uplus h)) \in [\![\mathbb{C}]\!]\epsilon$.

Proof. Follows by straightforward induction on the structure of $\mathbb{C}$.

Corollary 1. *For all* $s_1, s_2, h, h_1, h_2, \mathbb{C}, \epsilon$*, if* $((s_1, h_1), (s_2, h_2)) \in \texttt{foot}\,(\mathbb{C})\,\epsilon$ *and* $h_2 \# h$*, then* $((s_1, h_1 \uplus h), (s_2, h_2 \uplus h)) \in [\![\mathbb{C}]\!]\epsilon$.

Proof. Follows immediately from Lemma 2 and Lemma 4.

Theorem 5. *For all* $\mathbb{C} \in \textsc{Comm}$ *and* $\epsilon \in \textsc{Exit}$*:* $[\![\mathbb{C}]\!]\epsilon = \texttt{frame}\,(\texttt{foot}\,(\mathbb{C})\,\epsilon)$.

Proof. By induction on the structure of $\mathbb{C}$.

Case $\mathbb{C} = \texttt{skip}$
There are two cases to consider: 1) $\epsilon{=}er(\textsc{l}')$ for some $\textsc{l}'$; or 2) $\epsilon{=}ok$. In case (1) by definition we have $[\![\mathbb{C}]\!]\epsilon{=}\emptyset{=}\texttt{frame}\,(\texttt{foot}\,(\mathbb{C})\,\epsilon)$, as required. In case (2), from the definitions of $\texttt{foot}\,(.)$, $\texttt{frame}\,(.)$ and $[\![.]\!]$ we have $[\![\mathbb{C}]\!]\epsilon = \big\{(\sigma, \sigma) \,\big|\, \sigma \in \textsc{State}\big\} = \big\{\big((s, h_0 \uplus h), (s, h_0 \uplus h)\big) \,\big|\, h \in \textsc{Heap}\big\} {=}\texttt{frame}\,(\texttt{foot}\,(\mathbb{C})\,\epsilon)$, as required.

Case $\mathbb{C} = x{:=}\ e$
There are two cases to consider: 1) $\epsilon{=}er(\textsc{l}')$ for some $\textsc{l}'$; or 2) $\epsilon{=}ok$. In case (1) by definition we have $[\![\mathbb{C}]\!]\epsilon{=}\emptyset{=}\texttt{frame}\,(\texttt{foot}\,(\mathbb{C})\,\epsilon)$, as required. In case (2) from the definitions of $\texttt{foot}\,(.)$, $\texttt{frame}\,(.)$ and $[\![.]\!]$ we have:

$$\begin{aligned} [\![\mathbb{C}]\!]\epsilon &= \big\{((s, h), (s[x \mapsto s(e)], h)) \,\big|\, (s, h) \in \textsc{State}\big\} \\ &= \big\{((s, h_0 \uplus h), (s[x \mapsto s(e)], h_0 \uplus h)) \,\big|\, h \in \textsc{Heap}\big\} \\ &= \texttt{frame}\,(\texttt{foot}\,(\mathbb{C})\,\epsilon) \end{aligned}$$

Case $\mathbb{C} = x{:=}\ \texttt{*}$
The proof of this case is analogous to that of the previous case and is omitted.

Case $\mathbb{C} = \textsc{l}\!: x := [y]$
There are three cases to consider: 1) $\epsilon{=}ok$; or 2) $\epsilon{=}er(\textsc{l})$; or 3) $\epsilon{=}er(\textsc{l}')$ for some $\textsc{l}' \neq \textsc{l}$. In case (1) from the definitions of $\texttt{foot}\,(.)$, $\texttt{frame}\,(.)$ and $[\![.]\!]$ we have:

$$
\begin{aligned}
[\![\mathbb{C}]\!]\epsilon &= \big\{((s,h),(s[x\mapsto v],h)) \,\big|\, h(s(y)){=}v \in \textsc{Val}\big\}\\
&= \big\{((s,[l\mapsto v]\uplus h),(s[x\mapsto v],[l\mapsto v]\uplus h)) \,\big|\, v{\in}\textsc{Val} \wedge s(y){=}l \wedge h{\in}\textsc{Heap}\big\}\\
&= \texttt{frame}\,(\texttt{foot}\,(\mathbb{C})\,\epsilon)
\end{aligned}
$$

Similarly, in case (2) we have:

$$
\begin{aligned}
[\![\mathbb{C}]\!]\epsilon &= \big\{((s,h),(s,h)) \,\big|\, h(s(y)){=}\bot\big\}\\
&= \big\{((s,[l\mapsto \bot]\uplus h),(s,[l\mapsto \bot]\uplus h)) \,\big|\, s(y){=}l \wedge h{\in}\textsc{Heap}\big\}\\
&= \texttt{frame}\,(\texttt{foot}\,(\mathbb{C})\,\epsilon)
\end{aligned}
$$

Finally, in case (3) we have $[\![\mathbb{C}]\!]\epsilon{=}\emptyset{=}\texttt{frame}\,(\texttt{foot}\,(\mathbb{C})\,\epsilon)$, as required.

Case $\mathbb{C} = \textsc{l}\!: [x] := y$
The proof of this case is analogous to that of the previous case and is omitted.

Case $\mathbb{C} = x := \texttt{alloc()}$
There are two cases to consider: 1) $\epsilon{=}er(\textsc{l}')$ for some $\textsc{l}'$; or 2) $\epsilon{=}ok$. In case (1) by definition we have $[\![\mathbb{C}]\!]\epsilon{=}\emptyset{=}\texttt{frame}\,(\texttt{foot}\,(\mathbb{C})\,\epsilon)$, as required. In case (2) from the definitions of $\texttt{foot}\,(.)$, $\texttt{frame}\,(.)$ and $[\![.]\!]$ we have:

$$
\begin{aligned}
[\![\mathbb{C}]\!]\epsilon &= \big\{((s,h),(s[x\mapsto l],h[l\mapsto v])) \,\big|\, v\in\textsc{Val} \wedge (l\notin dom(h) \vee h(l){=}\bot)\big\}\\
&= \left\{((s,h\uplus h'),(s[x\mapsto l],[l\mapsto v]\uplus h')) \,\middle|\, \begin{array}{l} v\in\textsc{Val} \wedge (h{=}h_0 \vee h{=}[l\mapsto\bot])\\ \wedge\ h'\in\textsc{Heap}\end{array}\right\}\\
&= \texttt{frame}\,(\texttt{foot}\,(\mathbb{C})\,\epsilon)
\end{aligned}
$$

Case $\mathbb{C} = \texttt{free}(x)$
There are three cases to consider: 1) $\epsilon{=}ok$; or 2) $\epsilon{=}er(\textsc{l})$; or 3) $\epsilon{=}er(\textsc{l}')$ for some $\textsc{l}' \neq \textsc{l}$. In case (1) from the definitions of $\texttt{foot}\,(.)$, $\texttt{frame}\,(.)$ and $[\![.]\!]$ we have:

$$
\begin{aligned}
[\![\mathbb{C}]\!]\epsilon &= \big\{((s,h),(s,h[l\mapsto\bot])) \,\big|\, s(x){=}l \wedge h(l)\in\textsc{Val}\big\}\\
&= \big\{((s,[l\mapsto v]\uplus h),(s,[l\mapsto\bot]\uplus h)) \,\big|\, s(x){=}l \wedge v{\in}\textsc{Val} \wedge h{\in}\textsc{Heap}\big\}\\
&= \texttt{frame}\,(\texttt{foot}\,(\mathbb{C})\,\epsilon)
\end{aligned}
$$

Similarly, in case (2) we have:

$$
\begin{aligned}
[\![\mathbb{C}]\!]\epsilon &= \big\{((s,h),(s,h)) \,\big|\, s(x){=}l \wedge h(l){=}\bot\big\}\\
&= \big\{((s,[l\mapsto\bot]\uplus h),(s,[l\mapsto\bot]\uplus h)) \,\big|\, s(x){=}l \wedge h{\in}\textsc{Heap}\big\}\\
&= \texttt{frame}\,(\texttt{foot}\,(\mathbb{C})\,\epsilon)
\end{aligned}
$$

Finally, in case (3) we have $[\![\mathbb{C}]\!]\epsilon{=}\emptyset{=}\texttt{frame}\,(\texttt{foot}\,(\mathbb{C})\,\epsilon)$, as required.

Case $\mathbb{C} = \textsc{l}\!: \texttt{error}$
There are three cases to consider: 1) $\epsilon{=}ok$; or 2) $\epsilon{=}er(\textsc{l}')$ for $\textsc{l}' \neq \textsc{l}$; or 3) $\epsilon{=}er(\textsc{l})$. In (1) and (2) by definition we have $[\![\mathbb{C}]\!]\epsilon{=}\emptyset{=}\texttt{frame}\,(\texttt{foot}\,(\mathbb{C})\,\epsilon)$, as

required. In (3) from the definitions of $\texttt{foot}\,(.)$, $\texttt{frame}\,(.)$ and $[\![.]\!]$ we have $[\![\mathbb{C}]\!]\epsilon = \big\{(\sigma,\sigma) \,\big|\, \sigma \in \textsc{State}\big\} = \big\{\big((s, h_0 \uplus h), (s, h_0 \uplus h)\big) \,\big|\, h \in \textsc{Heap}\big\} = \texttt{frame}\,(\texttt{foot}\,(\mathbb{C})\,\epsilon)$, as required.

Case $\mathbb{C} = \texttt{assume}(B)$
There are two cases to consider: 1) $\epsilon{=}er(\textsc{l}')$ for some $\textsc{l}'$; or 2) $\epsilon{=}ok$. In case (1) by definition we have $[\![\mathbb{C}]\!]\epsilon{=}\emptyset{=}\texttt{frame}\,(\texttt{foot}\,(\mathbb{C})\,\epsilon)$, as required. In case (2) from the definitions of $\texttt{foot}\,(.)$, $\texttt{frame}\,(.)$ and $[\![.]\!]$ we have:

$$\begin{aligned}
[\![\mathbb{C}]\!]\epsilon &= \big\{((s,h),(s,h)) \,\big|\, s(B) \neq 0\big\} \\
&= \big\{((s, h_0 \uplus h), (s, h_0 \uplus h)) \,\big|\, s(B) \neq 0 \land h \in \textsc{Heap}\big\} \\
&= \texttt{frame}\,(\texttt{foot}\,(\mathbb{C})\,\epsilon)
\end{aligned}$$

Case $\mathbb{C} = \texttt{local}\ x\ \texttt{in}\ \mathbb{C}'$

$$\forall \epsilon.\ [\![\mathbb{C}']\!]\epsilon{=}\texttt{frame}\,(\texttt{foot}\,(\mathbb{C}')\,\epsilon) \tag{I.H}$$

From the definitions of $\texttt{foot}\,(.)$, $\texttt{frame}\,(.)$, $[\![.]\!]$ and (I.H) we have:

$$\begin{aligned}
[\![\mathbb{C}]\!]\epsilon &\triangleq \left\{((s[x \mapsto v], h), (s'[x \mapsto v], h')) \,\middle|\, \begin{array}{l} ((s,h),(s',h')) \in [\![\mathbb{C}']\!]\epsilon \\ \land\ v \in \textsc{Val} \end{array}\right\} \\
&\overset{\text{(I.H)}}{=} \left\{((s[x \mapsto v], h), (s'[x \mapsto v], h')) \,\middle|\, \begin{array}{l} ((s,h),(s',h')) \in \texttt{frame}\,(\texttt{foot}\,(\mathbb{C}')\,\epsilon) \\ \land\ v \in \textsc{Val} \end{array}\right\} \\
&= \texttt{frame}\left(\left\{((s[x \mapsto v], h), (s'[x \mapsto v], h')) \,\middle|\, \begin{array}{l} ((s,h),(s',h')) \in \texttt{foot}\,(\mathbb{C}')\,\epsilon \\ \land\ v \in \textsc{Val} \end{array}\right\}\right) \\
&= \texttt{frame}\,(\texttt{foot}\,(\mathbb{C})\,\epsilon)
\end{aligned}$$

Case $\mathbb{C} = \mathbb{C}_1; \mathbb{C}_2$

$$\forall \epsilon.\ [\![\mathbb{C}_1]\!]\epsilon{=}\texttt{frame}\,(\texttt{foot}\,(\mathbb{C}_1)\,\epsilon) \land [\![\mathbb{C}_2]\!]\epsilon{=}\texttt{frame}\,(\texttt{foot}\,(\mathbb{C}_2)\,\epsilon) \tag{I.H}$$

In what follows we show $[\![\mathbb{C}]\!]\epsilon \subseteq \texttt{frame}\,(\texttt{foot}\,(\mathbb{C})\,\epsilon)$ and $\texttt{frame}\,(\texttt{foot}\,(\mathbb{C})\,\epsilon) \subseteq [\![\mathbb{C}]\!]\epsilon$, thus establishing $[\![\mathbb{C}]\!]\epsilon = \texttt{frame}\,(\texttt{foot}\,(\mathbb{C})\,\epsilon)$, as required.

For the first part, from the definitions of $\texttt{foot}\,(.)$, $\texttt{frame}\,(.)$, $[\![.]\!]$, Lemma 3 and (I.H) we have:

$$\begin{aligned}
[\![\mathbb{C}]\!]\epsilon &\triangleq \left\{(\sigma,\sigma') \,\middle|\, \begin{array}{l} \epsilon \neq ok \land (\sigma,\sigma') \in [\![\mathbb{C}_1]\!]\epsilon \\ \lor\ \exists \sigma''.\ (\sigma,\sigma'') \in [\![\mathbb{C}_1]\!]ok \land (\sigma'',\sigma') \in [\![\mathbb{C}_2]\!]\epsilon \end{array}\right\} \\
&= \big\{(\sigma,\sigma') \,\big|\, \epsilon \neq ok \land (\sigma,\sigma') \in [\![\mathbb{C}_1]\!]\epsilon\big\} \\
&\quad \cup \big\{(\sigma,\sigma') \,\big|\, \exists \sigma''.\ (\sigma,\sigma'') \in [\![\mathbb{C}_1]\!]ok \land (\sigma'',\sigma') \in [\![\mathbb{C}_2]\!]\epsilon\big\} \\
&\overset{I.H}{=} \big\{(\sigma,\sigma') \,\big|\, \epsilon \neq ok \land (\sigma,\sigma') \in \texttt{frame}\,(\texttt{foot}\,(\mathbb{C}_1)\,\epsilon)\big\} \\
&\quad \cup \left\{(\sigma,\sigma') \,\middle|\, \begin{array}{l} \exists \sigma''.\ (\sigma,\sigma'') \in \texttt{frame}\,(\texttt{foot}\,(\mathbb{C}_1)\,ok) \\ \qquad \land\ (\sigma'',\sigma') \in \texttt{frame}\,(\texttt{foot}\,(\mathbb{C}_2)\,\epsilon) \end{array}\right\} \\
&= \texttt{frame}\,\big(\big\{(\sigma_1,\sigma_2) \,\big|\, \epsilon \neq ok \land (\sigma_1,\sigma_2) \in \texttt{foot}\,(\mathbb{C}_1)\,\epsilon\big\}\big) \\
&\quad \cup \left\{((s_1, h_1 \uplus h), (s_2, h_2 \uplus h')) \,\middle|\, \begin{array}{l} \exists s'', h_3, h_4.\ ((s_1,h_1),(s',h_3)) \in \texttt{foot}\,(\mathbb{C}_1)\,ok \\ \qquad \land\ ((s',h_4),(s_2,h_2)) \in \texttt{foot}\,(\mathbb{C}_2)\,\epsilon \\ \qquad \land\ h_3 \uplus h{=}h_4 \uplus h' \end{array}\right\}
\end{aligned}$$

$$
\begin{aligned}
&(\textit{Lemma 3}) \subseteq \texttt{frame}\left(\left\{(\sigma_1, \sigma_2) \,\middle|\, \epsilon \neq \mathit{ok} \wedge (\sigma_1, \sigma_2) \in \texttt{foot}\,(\mathbb{C}_1)\,\epsilon\right\}\right) \\
&\qquad \cup \left\{ ((s_1, h_1 \uplus h), (s_2, h_2 \uplus h')) \,\middle|\, \begin{array}{l} \exists s'', h_3, h_4.\ ((s_1, h_1), (s', h_3)) \in \texttt{foot}\,(\mathbb{C}_1)\ \mathit{ok} \\ \qquad \wedge\ ((s', h_4), (s_2, h_2)) \in \texttt{foot}\,(\mathbb{C}_2)\,\epsilon \\ \wedge\ \exists h_{34}, h_{3b}, h_{a4}, h_{ab}. \\ \quad h_3{=}h_{34} \uplus h_{3b} \wedge h{=}h_{a4} \uplus h_{ab} \\ \quad \wedge\ h_4{=}h_{34} \uplus h_{a4} \wedge h'{=}h_{3b} \uplus h_{ab} \end{array} \right\} \\
&= \texttt{frame}\left(\left\{(\sigma_1, \sigma_2) \,\middle|\, \epsilon \neq \mathit{ok} \wedge (\sigma_1, \sigma_2) \in \texttt{foot}\,(\mathbb{C}_1)\,\epsilon\right\}\right) \\
&\qquad \cup \left\{ ((s_1, h_1 \uplus h_{a4} \uplus h_{ab}), (s_2, h_2 \uplus h_{3b} \uplus h_{ab}) \,\middle|\, \begin{array}{l} \exists h_{34}.\ ((s_1, h_1), (s', h_{34} \uplus h_{3b})) \in \texttt{foot}\,(\mathbb{C}_1)\ \mathit{ok} \\ \qquad \wedge\ ((s', h_{34} \uplus h_{a4}), (s_2, h_2)) \in \texttt{foot}\,(\mathbb{C}_2)\,\epsilon \end{array} \right\} \\
&= \texttt{frame}\left(\left\{(\sigma_1, \sigma_2) \,\middle|\, \epsilon \neq \mathit{ok} \wedge (\sigma_1, \sigma_2) \in \texttt{foot}\,(\mathbb{C}_1)\,\epsilon\right\}\right) \\
&\qquad \cup \texttt{frame}\left(\left\{ ((s_1, h_1 \uplus h_{a4}), (s_2, h_2 \uplus h_{3b})) \,\middle|\, \begin{array}{l} \exists h_{34}.\ ((s_1, h_1), (s', h_{34} \uplus h_{3b})) \in \texttt{foot}\,(\mathbb{C}_1)\ \mathit{ok} \\ \qquad \wedge\ ((s', h_{34} \uplus h_{a4}), (s_2, h_2)) \in \texttt{foot}\,(\mathbb{C}_2)\,\epsilon \end{array} \right\} \right) \\
&= \texttt{frame}\left(\begin{array}{l} \left\{(\sigma_1, \sigma_2) \,\middle|\, \epsilon \neq \mathit{ok} \wedge (\sigma_1, \sigma_2) \in \texttt{foot}\,(\mathbb{C}_1)\,\epsilon\right\} \\ \cup \left\{ (\sigma_1 \bullet \sigma_{a4}, \sigma_2 \bullet \sigma_{3b}) \,\middle|\, \begin{array}{l} \exists \sigma_{34}.\ (\sigma_1, \sigma_{34} \bullet \sigma_{3b}) \in \texttt{foot}\,(\mathbb{C}_1)\ \mathit{ok} \\ \qquad \wedge\ (\sigma_{34} \bullet \sigma_{a4}, \sigma_2) \in \texttt{foot}\,(\mathbb{C}_2)\,\epsilon \end{array} \right\} \end{array} \right) \\
&= \texttt{frame}\,(\texttt{foot}\,(\mathbb{C}_1; \mathbb{C}_2)\,\epsilon) \\
&= \texttt{frame}\,(\texttt{foot}\,(\mathbb{C})\,\epsilon)
\end{aligned}
$$

For the second part, from the definitions of $\texttt{foot}\,(.)$, $\texttt{frame}\,(.)$, $[\![.]\!]$, Lemma 4 and (I.H) we have:

$$
\begin{aligned}
&\texttt{frame}\,(\texttt{foot}\,(\mathbb{C})\,\epsilon) = \left\{((s_1, h_1 \uplus h_r), (s_2, h_2 \uplus h_r)) \,\middle|\, ((s_1, h_1), (s_2, h_2)) \in \texttt{foot}\,(\mathbb{C})\,\epsilon \wedge h \in \textsc{Heap}\right\} \\
&\quad = \left\{((s_1, h_1 \uplus h_r), (s_2, h_2 \uplus h_r)) \,\middle|\, \epsilon \neq \mathit{ok} \wedge ((s_1, h_1), (s_2, h_2)) \in \texttt{foot}\,(\mathbb{C}_1)\,\epsilon\right\} \\
&\qquad \cup \left\{ ((s_1, h_1 \uplus h \uplus h_r), (s_2, h_2 \uplus h' \uplus h_r)) \,\middle|\, \begin{array}{l} \exists h_c, s_3.\ ((s_1, h_1), (s_3, h' \uplus h_c)) \in \texttt{foot}\,(\mathbb{C}_1)\ \mathit{ok} \\ \qquad \wedge\ ((s_3, h_c \uplus h), (s_2, h_2)) \in \texttt{foot}\,(\mathbb{C}_2)\,\epsilon \end{array} \right\} \\
&(\textit{Corollary 1}) \subseteq \left\{((s_1, h_1 \uplus h_r), (s_2, h_2 \uplus h_r)) \,\middle|\, \epsilon \neq \mathit{ok} \wedge ((s_1, h_1 \uplus h_r), (s_2, h_2 \uplus h_r)) \in [\![\mathbb{C}_1]\!]\epsilon\right\} \\
&\qquad \cup \left\{ ((s_1, h_1 \uplus h \uplus h_r), (s_2, h_2 \uplus h' \uplus h_r)) \,\middle|\, \begin{array}{l} \exists h_c, s_3. \\ (((s_1, h_1 \uplus h \uplus h_r), (s_3, h' \uplus h_c \uplus h \uplus h_r)) \in [\![\mathbb{C}_1]\!]\mathit{ok} \\ \wedge\ ((s_3, h_c \uplus h \uplus h' \uplus h_r), (s_2, h_2 \uplus h' \uplus h_r)) \in [\![\mathbb{C}_2]\!]\epsilon \end{array} \right\} \\
&\quad = \left\{(\sigma, \sigma') \,\middle|\, \epsilon \neq \mathit{ok} \wedge (\sigma, \sigma') \in [\![\mathbb{C}_1]\!]\epsilon\right\} \\
&\qquad \cup \left\{(\sigma, \sigma') \,\middle|\, \exists \sigma''.\ (\sigma, \sigma'') \in [\![\mathbb{C}_1]\!]\mathit{ok} \wedge (\sigma'', \sigma') \in [\![\mathbb{C}_2]\!]\epsilon\right\} \\
&\quad = \left\{ (\sigma, \sigma') \,\middle|\, \begin{array}{l} \epsilon \neq \mathit{ok} \wedge (\sigma, \sigma') \in [\![\mathbb{C}_1]\!]\epsilon \\ \vee\ \exists \sigma''.\ (\sigma, \sigma'') \in [\![\mathbb{C}_1]\!]\mathit{ok} \wedge (\sigma'', \sigma') \in [\![\mathbb{C}_2]\!]\epsilon \end{array} \right\} \\
&\quad = [\![\mathbb{C}]\!]\epsilon
\end{aligned}
$$

Case $\mathbb{C} = \mathbb{C}_1 + \mathbb{C}_2$

$$\forall \epsilon.\ [\![\mathbb{C}_1]\!]\epsilon{=}\texttt{frame}\,(\texttt{foot}\,(\mathbb{C}_1)\,\epsilon) \wedge [\![\mathbb{C}_2]\!]\epsilon{=}\texttt{frame}\,(\texttt{foot}\,(\mathbb{C}_2)\,\epsilon) \qquad \text{(I.H)}$$

From the definitions of $\texttt{foot}\,(.)$, $\texttt{frame}\,(.)$, $[\![.]\!]$ and (I.H) we have:

$$
\begin{aligned}
[\![\mathbb{C}]\!]\epsilon &= [\![\mathbb{C}_1]\!]\epsilon \cup [\![\mathbb{C}_2]\!]\epsilon \\
&= \texttt{frame}\,(\texttt{foot}\,(\mathbb{C}_1)\,\epsilon) \cup \texttt{frame}\,(\texttt{foot}\,(\mathbb{C}_2)\,\epsilon)
\end{aligned}
$$

$$= \texttt{frame}\,(\texttt{foot}\,(\mathbb{C}_1 + \mathbb{C}_2)\,\epsilon)$$
$$= \texttt{frame}\,(\texttt{foot}\,(\mathbb{C})\,\epsilon)$$

Case $\mathbb{C} = \mathbb{C}_r^\star$
We first demonstrate that:

$$\forall i \in \mathbb{N}.\ [\![\mathbb{C}_r^i]\!]\epsilon{=}\texttt{frame}\left(\texttt{foot}\left(\mathbb{C}_r^i\right)\epsilon\right) \tag{1}$$

We proceed by induction on i.
Base case $i = 0$
It suffices to show that $[\![\texttt{skip}]\!]\epsilon{=}\texttt{frame}\,(\texttt{foot}\,(\texttt{skip})\,\epsilon)$, which follows from the proof of case `skip` above.

Inductive case $i = n{+}1$
From the definition of $\mathbb{C}_r^{n+1}$ we then have $[\![\mathbb{C}_r^{n+1}]\!]\epsilon = [\![\mathbb{C}_r; \mathbb{C}_r^n]\!]\epsilon$. On the other hand, from the proof of the sequential case composition we have $[\![\mathbb{C}_r; \mathbb{C}_r^n]\!]\epsilon = \texttt{frame}\,(\texttt{foot}\,(\mathbb{C}_r; \mathbb{C}_r^n)\,\epsilon)$, and thus we have $[\![\mathbb{C}_r^{n+1}]\!]\epsilon = \texttt{frame}\,(\texttt{foot}\,(\mathbb{C}_r; \mathbb{C}_r^n)\,\epsilon)$. Finally, from definition of $\mathbb{C}_r^{n+1}$ we have $\mathbb{C}_r; \mathbb{C}_r^n = \mathbb{C}_r^{n+1}$ and thus we have $[\![\mathbb{C}_r^{n+1}]\!]\epsilon = \texttt{frame}\left(\texttt{foot}\left(\mathbb{C}_r^{n+1}\right)\epsilon\right)$, as required.

From the definitions of $\texttt{foot}\,(.)$, $\texttt{frame}\,(.)$, $[\![.]\!]$ and (1) we have:

$$\begin{aligned}
[\![\mathbb{C}]\!]\epsilon &= \bigcup_{i\in\mathbb{N}} [\![\mathbb{C}_r^i]\!]\epsilon \\
&\overset{(1)}{=} \bigcup_{i\in\mathbb{N}} \texttt{frame}\left(\texttt{foot}\left(\mathbb{C}_r^i\right)\epsilon\right) \\
&= \texttt{frame}\left(\bigcup_{i\in\mathbb{N}} \texttt{foot}\left(\mathbb{C}_r^i\right)\epsilon\right) \\
&= \texttt{frame}\,(\texttt{foot}\,(\mathbb{C})\,\epsilon)
\end{aligned}$$

D Symbolic Execution Rules

We now list all rules for the analysis described in §5.

SE-Seq1
$$\frac{[p]\,\mathbb{C}\,[ok\colon q]\ \mathbb{C}_i \rightsquigarrow [p_1]\ \mathbb{C};\mathbb{C}_1\ [\epsilon_1 : q_1] \qquad [p_1]\,\mathbb{C};\mathbb{C}_1\,[\epsilon_1 : q_1]\ \mathbb{C}_2 \rightsquigarrow [p_2]\ \mathbb{C};\mathbb{C}_1;\mathbb{C}_2\ [\epsilon_2 : q_2]}{[p]\,\mathbb{C}\,[ok\colon q]\ \mathbb{C}_1;\mathbb{C}_2 \rightsquigarrow [p_2]\ \mathbb{C};\mathbb{C}_1;\mathbb{C}_2\ [\epsilon_2 : q_2]}$$

SE-Seq2
$$[p]\,\mathbb{C}_1\,[er(\mathrm{L})\colon q]\ \mathbb{C}_2 \rightsquigarrow [p]\ \mathbb{C}_1;\mathbb{C}_2\ [er(\mathrm{L})\colon q]$$

SE-Local
$$\frac{[p[x'/x]]\,\mathbb{C}\,[ok\colon q[x'/x]]\ \mathbb{C}' \rightsquigarrow [p']\ \mathbb{C};\mathbb{C}'\ [\epsilon : q'] \qquad x' \notin \mathsf{fv}(p,q,\mathbb{C},\mathbb{C}') \qquad z \notin \mathsf{fv}(p',q',\mathbb{C},\mathbb{C}')}{[p]\,\mathbb{C}\,[ok\colon q]\ \texttt{local}\ x\ \texttt{in}\ \mathbb{C}' \rightsquigarrow [p'[z/x']]\ \mathbb{C};\texttt{local}\ x\ \texttt{in}\ \mathbb{C}'\ [\epsilon : q'[z/x']]}$$

SE-Loop
$$\frac{[p]\,\mathbb{C}'\,[ok\colon q]\ \texttt{skip} + \mathbb{C} + (\mathbb{C};\mathbb{C}) + \cdots + \mathbb{C}^{N_{loops}} \rightsquigarrow [p']\ \mathbb{C}';\texttt{skip} + \mathbb{C} + (\mathbb{C};\mathbb{C}) + \cdots + \mathbb{C}^{N_{loops}}\ [\epsilon : q']}{[p]\,\mathbb{C}'\,[ok\colon q]\ \mathbb{C}^\star \rightsquigarrow [p']\ \mathbb{C}';\mathbb{C}^\star\ [\epsilon : q']}$$

SE-Choice
$$\frac{[p]\,\mathbb{C}\,[ok\colon q]\ \mathbb{C}_i \rightsquigarrow [p_i]\ \mathbb{C};\mathbb{C}_i\ [\epsilon_i : q_i]}{[p]\,\mathbb{C}\,[ok\colon q]\ \mathbb{C}_1 + \mathbb{C}_2 \rightsquigarrow [p_i]\ \mathbb{C};\mathbb{C}_1 + \mathbb{C}_2\ [\epsilon_i : q_i]}$$

SE-Skip
$$[p]\,\mathbb{C}\,[ok\colon q]\ \texttt{skip} \rightsquigarrow [p]\,\mathbb{C};\texttt{skip}\,[ok\colon q]$$

SE-Error
$$[p]\,\mathbb{C}\,[ok\colon q]\ \mathrm{L}\colon\texttt{error} \rightsquigarrow [p]\ \mathbb{C};\mathrm{L}\colon\texttt{error}\ [er(\mathrm{L})\colon q]$$

SE-Assign
$$\frac{x' \notin \mathsf{fv}(p,\mathbb{C},x,e,q)}{[p]\,\mathbb{C}\,[ok\colon q]\ x := e \rightsquigarrow [p]\ \mathbb{C};x := e\,[ok\colon x = e[x'/x] * q[x'/x]]}$$

SE-Havoc
$$\frac{x' \notin \mathsf{fv}(p,\mathbb{C},x,q)}{[p]\,\mathbb{C}\,[ok\colon q]\ x := * \rightsquigarrow [p]\ \mathbb{C};x := *\,[ok\colon q[x'/x]]}$$

SE-Assume
$$[p]\,\mathbb{C}\,[ok\colon q]\ \texttt{assume}(B) \rightsquigarrow [p]\ \mathbb{C};\texttt{assume}(B)\ [ok\colon B * q]$$

Note that SE-Assume does not require $B * q$ to be explicitly satisfiable. This is because we implicitly stop the symbolic execution any time either the inferred presumption or current state becomes inconsistent due to the application of any rule.

SE-Alloc1
$$\frac{x', v \notin \mathsf{fv}(p,\mathbb{C},x,q)}{[p]\,\mathbb{C}\,[ok\colon q]\ x := \texttt{alloc()} \rightsquigarrow [p]\ \mathbb{C};x := \texttt{alloc()}\ [ok\colon q[x'/x] * x \mapsto v]}$$

There is no rule corresponding to Alloc2 in our analysis. This is not a fundamental choice but rather a practical one, as including such a rule would introduce branching on all known invalidated addresses at each `alloc()` call site, which can blow up the exploration space. To put it another way, we could easily include an analogue of SE-Alloc1 for re-using known-invalidated addresses; the ability not to do so is granted to us by the under-approximate setting.

SE-Load

$$\frac{y \mapsto e * F \vdash q * M \qquad \mathsf{mod}(\mathbb{C}) \cap \mathsf{fv}(M) = \emptyset \qquad x \notin \mathsf{fv}(F) \qquad x' \notin \mathsf{fv}(p, \mathbb{C}, x, y, q)}{[p]\, \mathbb{C}\, [ok\colon q]\ x := [y] \rightsquigarrow [p * M]\, \mathbb{C}; x := [y]\, [ok\colon y \mapsto e[x'/x] * x = e[x'/x] * F]}$$

SE-LoadEr

$$\frac{q \vdash y \not\mapsto\ * \mathsf{true}}{[p]\, \mathbb{C}\, [ok\colon q]\ \textsc{l}\colon x := [y] \rightsquigarrow [p]\ \mathbb{C}; \textsc{l}\colon x := [y]\ [er(\textsc{l})\colon q]}$$

SE-LoadNull

$$\frac{q \vdash y = \mathtt{null} * \mathsf{true}}{[p]\, \mathbb{C}\, [ok\colon q]\ \textsc{l}\colon x := [y] \rightsquigarrow [p]\ \mathbb{C}; \textsc{l}\colon x := [y]\ [er(\textsc{l})\colon q]}$$

SE-Store

$$\frac{x \mapsto e * F \vdash q * M \qquad \mathsf{mod}(\mathbb{C}) \cap \mathsf{fv}(M) = \emptyset}{[p]\, \mathbb{C}\, [ok\colon q]\ [x] := y \rightsquigarrow [p * M]\, \mathbb{C}; [x] := y\, [ok\colon x \mapsto y * F]}$$

SE-StoreEr

$$\frac{q \vdash x \not\mapsto\ * \mathsf{true}}{[p]\, \mathbb{C}\, [ok\colon q]\ \textsc{l}\colon [x] := y \rightsquigarrow [p]\ \mathbb{C}; \textsc{l}\colon [x] := y\ [er(\textsc{l})\colon q]}$$

SE-StoreNull

$$\frac{q \vdash x = \mathtt{null} * \mathsf{true}}{[p]\, \mathbb{C}\, [ok\colon q]\ \textsc{l}\colon [x] := y \rightsquigarrow [p]\ \mathbb{C}; \textsc{l}\colon [x] := y\ [er(\textsc{l})\colon q]}$$

SE-Free

$$\frac{x \mapsto e * F \vdash q * M \qquad \mathsf{mod}(\mathbb{C}) \cap \mathsf{fv}(M) = \emptyset}{[p]\, \mathbb{C}\, [ok\colon q]\ \mathtt{free}(x) \rightsquigarrow [p * M]\, \mathbb{C}; \mathtt{free}(x)\, [ok\colon x \not\mapsto\ * F]}$$

SE-FreeEr

$$\frac{q \vdash x \not\mapsto\ * \mathsf{true}}{[p]\, \mathbb{C}\, [ok\colon q]\ \mathtt{free}(x) \rightsquigarrow [p]\ \mathbb{C}; \mathtt{free}(x)\ [er(\textsc{l})\colon q]}$$