

Separation Logic in the Presence of Garbage Collection

Technical Appendix

Chung-Kil Hur Derek Dreyer Viktor Vafeiadis

Max Planck Institute for Software Systems (MPI-SWS)

`{gil,dreyer,viktor}@mpi-sws.org`

April 2011

Contents

1	Language	4
1.1	Storage Model	4
1.2	Syntax	4
1.3	Operational Semantics	5
1.4	Garbage Collector Specification	6
2	Program Specifications	7
2.1	Logical Storage Model	7
2.2	Syntax	8
2.3	Semantics	9
3	Program Logic	12
3.1	Inner-level rules	12
3.2	Outer-level rules	13
3.3	Assertion entailments	14
3.4	Derived rules	14
3.5	Problematic rules	15
4	Examples	16
4.1	Array Assignment	16
4.2	Word Swap	16
4.3	Linking of Assignment and Swap	17
4.4	Simple Addition	17
4.5	Integer Arithmetic	17
4.6	List Reversal	18
4.7	Array Copy	19
5	Soundness of Program Logic	22
5.1	Basic Lemmas	22
5.2	Soundness of Inner-level Rules	25
5.2.1	Skip	25
5.2.2	Assign	25
5.2.3	Read	26
5.2.4	Write	27
5.2.5	Seq	29
5.2.6	Frame	31
5.2.7	Conseq	32
5.2.8	Ex	33
5.2.9	Gen	34
5.2.10	Total	34
5.2.11	If	34
5.2.12	While	35
5.3	Soundness of Outer-level Rules	38
5.3.1	Alloc	38
5.3.2	Incl	42

5.3.3	Seq	44
5.3.4	Frame	46
5.3.5	Conseq	47
5.3.6	Ex	47
5.3.7	Gen	48
5.3.8	Total	48
5.3.9	If	49
5.3.10	While	50
5.4	Soundness of Assertion Entailments	53
5.4.1	NPtrSafe	53
5.4.2	BoolWord	53
5.4.3	PointstoNZero	53
5.4.4	ExpSafe	54
5.4.5	HeapSafe	54
5.4.6	ExpHeapSafe	54
5.4.7	SafeEq	54
5.5	Soundness of Derived Rules	55
5.5.1	Ex'	55
5.5.2	Disj	55
5.5.3	Inst	55
5.5.4	Assign'	56
5.5.5	Read' and Read''	57
5.5.6	ASSIGN and ASSIGN'	57
5.5.7	READ and READ'	58
5.5.8	WRITE and WRITE'	59
5.5.9	ALLOC	59

1 Language

1.1 Storage Model

$$\begin{aligned}\text{ProgVars} &\stackrel{\text{def}}{=} \{x, y, \dots\} \\ \text{Words} &\stackrel{\text{def}}{=} \{w \in \mathbb{Z}\} \\ \text{Ptrs} &\stackrel{\text{def}}{=} \{p \in \text{Words} \mid p > 0 \wedge p \text{ is a multiple of } 4\} \\ \text{NonPtrs} &\stackrel{\text{def}}{=} \{a \in \text{Words} \setminus \text{Ptrs}\} \\ \text{Stores} &\stackrel{\text{def}}{=} \{s \in \text{ProgVars} \rightarrow \text{Words}\} \\ \text{Heaps} &\stackrel{\text{def}}{=} \{h \in \text{Ptrs} \rightarrow_{\text{fin}} \text{Words}\}\end{aligned}$$

1.2 Syntax

Expressions

$$E \in \text{Exps} ::= \begin{array}{l} x \\ w \\ E \star E \\ \text{not } E \end{array}$$

where $x \in \text{ProgVars}$, $w \in \text{Words}$ and $\star \in \{+, -, \times, \div, <, =, \text{and}\}$

$$\begin{aligned}E_1 \text{ or } E_2 &\stackrel{\text{def}}{=} \text{not } (\text{not } E_1 \text{ and not } E_2) \\ E_1 \neq E_2 &\stackrel{\text{def}}{=} \text{not } (E_1 = E_2) \\ E_1 \leq E_2 &\stackrel{\text{def}}{=} (E_1 = E_2) \text{ or } (E_1 < E_2) \\ \text{ENC}(E) &\stackrel{\text{def}}{=} 2 \times E + 1\end{aligned}$$

Commands

$$C ::= \begin{array}{l} x := E \\ x := [E] \\ [E] := E \\ \text{skip} \\ \text{if } E \text{ then } C \text{ else } C \text{ fi} \\ \text{while } E \text{ do } C \text{ od} \\ C; C \\ \text{alloc } x \end{array}$$

where $x \in \text{ProgVars}$ and $i \in \mathbb{N}$

$$x := \text{ALLOC}(E) \stackrel{\text{def}}{=} x := E; \text{alloc } x$$

Free variables

$$\begin{aligned}
\text{FPV}(E) &\stackrel{\text{def}}{=} \text{the set of program variables appearing in the expression } E \\
\text{FPV}(C) &\stackrel{\text{def}}{=} \text{the set of program variables appearing in the command } C \\
\text{Mod}(C) &\stackrel{\text{def}}{=} \begin{cases} \{x\} & \text{if } C = (x := E) \vee C = (x := [E]) \vee C = \text{alloc } x \\ \text{Mod}(C') & \text{if } C = \text{while } E \text{ do } C' \text{ od} \\ \text{Mod}(C') \cup \text{Mod}(C'') & \text{if } C = \text{if } E \text{ then } C' \text{ else } C'' \text{ fi} \vee C = C'; C'' \\ \emptyset & \text{otherwise} \end{cases}
\end{aligned}$$

1.3 Operational Semantics

$\llbracket E \rrbracket \in \text{Stores} \rightarrow \text{Words}$

$$\begin{aligned}
\llbracket x \rrbracket_s &::= s(x) \\
\llbracket w \rrbracket_s &::= w \\
\llbracket E_1 \star E_2 \rrbracket_s &::= \begin{cases} w_1 \star w_2 & \text{if } \llbracket E_1 \rrbracket_s = w_1 \wedge \llbracket E_2 \rrbracket_s = w_2 \\ \text{undef} & \text{otherwise} \end{cases} \\
\text{where } \star \in \{+, -, \times, \div, <, =, \text{and}\}, & w \div 0 = \text{undef}, \\
w_1 < w_2 &\stackrel{\text{def}}{=} 1 \text{ if } w_1 < w_2; 0 \text{ otherwise,} \\
w_1 = w_2 &\stackrel{\text{def}}{=} 1 \text{ if } w_1 = w_2; 0 \text{ otherwise,} \\
w_1 \text{ and } w_2 &\stackrel{\text{def}}{=} 1 \text{ if } w_1 \neq 0 \wedge w_2 \neq 0; 0 \text{ otherwise} \\
\llbracket \text{not } E \rrbracket_s &::= \begin{cases} \text{not } w & \text{if } \llbracket E \rrbracket_s = w \\ \text{undef} & \text{otherwise} \end{cases} \\
\text{where not } w &\stackrel{\text{def}}{=} 1 \text{ if } w = 0; 0 \text{ otherwise}
\end{aligned}$$

$C, s, h \rightsquigarrow C', s', h'$

$$\begin{array}{c}
\frac{\llbracket E \rrbracket_s \neq \text{undef}}{x := E, s, h \rightsquigarrow \text{skip}, (s \mid x \mapsto \llbracket E \rrbracket_s), h} \\
\frac{\llbracket E \rrbracket_s = p \in \text{dom}(h)}{x := [E], s, h \rightsquigarrow \text{skip}, (s \mid x \mapsto h(p)), h} \\
\frac{\llbracket E \rrbracket_s = p \in \text{dom}(h) \quad \llbracket E' \rrbracket_s \neq \text{undef}}{[E] := E', s, h \rightsquigarrow \text{skip}, s, (h \mid p \mapsto \llbracket E' \rrbracket_s)} \\
\frac{\llbracket E \rrbracket_s \in \text{Words} \setminus \{0\}}{\text{if } E \text{ then } C_1 \text{ else } C_2 \text{ fi}, s, h \rightsquigarrow C_1, s, h} \\
\frac{\llbracket E \rrbracket_s = 0}{\text{if } E \text{ then } C_1 \text{ else } C_2 \text{ fi}, s, h \rightsquigarrow C_2, s, h} \\
\frac{\llbracket E \rrbracket_s = 0}{\text{while } E \text{ do } C \text{ od}, s, h \rightsquigarrow \text{skip}, s, h} \\
\frac{\llbracket E \rrbracket_s \in \text{Words} \setminus \{0\}}{\text{while } E \text{ do } C \text{ od}, s, h \rightsquigarrow (C; \text{while } E \text{ do } C \text{ od}), s, h} \\
\frac{C_1, s, h \rightsquigarrow C'_1, s', h'}{(\text{skip}; C), s, h \rightsquigarrow C, s, h} \quad \frac{C_1, s, h \rightsquigarrow C'_1, s', h'}{(C_1; C_2), s, h \rightsquigarrow (C'_1; C_2), s', h'}
\end{array}$$

Notation

$$C, s, h \rightsquigarrow - \quad \text{iff } \exists C', s', h'. C, s, h \rightsquigarrow C', s', h'$$

$$C, s, h \text{ diverges} \quad \text{iff } \exists \{C_i, s_i, h_i\}_{i \in \mathbb{N}}. (C_0, s_0, h_0) = (C, s, h) \wedge \forall i. C_i, s_i, h_i \rightsquigarrow C_{i+1}, s_{i+1}, h_{i+1}$$

1.4 Garbage Collector Specification

$$\text{Shapes} \quad := \{ \sigma \in \text{Ptrs} \rightarrow_{\text{fin}} \mathbb{N}^+ \}$$

$$\overline{\text{dom}}(\sigma) \quad := \bigsqcup_{p \in \text{dom}(\sigma)} \{ p + 4 \times 0, \dots, p + 4(\sigma(p) - 1) \}$$

$$\text{roots}(s) \quad \stackrel{\text{def}}{=} \{ p \in \text{Ptrs} \mid \exists \mathbf{x}. p = s(\mathbf{x}) \}$$

$$\text{reach}_0(R, h, \sigma) \quad \stackrel{\text{def}}{=} R$$

$$\text{reach}_{n+1}(R, h, \sigma) \quad \stackrel{\text{def}}{=} \text{reach}_n(R, h, \sigma) \cup \{ p \in \text{Ptrs} \mid \exists p' \in \text{reach}_n(R, h, \sigma). \exists i < \sigma(p'). p = h(p' + 4i) \}$$

$$\text{reach}(R, h, \sigma) \quad \stackrel{\text{def}}{=} \bigcup_{n \in \mathbb{N}} \text{reach}_n(R, h, \sigma)$$

$$(s, h, \sigma) \cong (s', h', \sigma') \quad \stackrel{\text{def}}{=} \exists r \in \text{Bij}(\text{reach}(\text{roots}(s), h, \sigma), \text{reach}(\text{roots}(s'), h', \sigma')). \\ (\forall \mathbf{x}. (s(\mathbf{x}), s'(\mathbf{x})) \in \bar{r}) \wedge \\ (\forall (p, p') \in r. \exists n. \sigma(p) = \sigma'(p') = n \wedge \forall i. 0 \leq i < n \implies (h(p + 4i), h'(p' + 4i)) \in \bar{r}) \\ \text{where } \bar{r} \stackrel{\text{def}}{=} r \cup \{ (a, a) \mid a \in \text{NonPtrs} \}$$

$$[p \mapsto_n w_0, \dots, w_{n-1}] \quad \stackrel{\text{def}}{=} (\emptyset \mid p + 4 \times 0 \mapsto w_0 \mid \dots \mid p + 4(n-1) \mapsto w_{n-1}) \in \text{Heaps}$$

$$[p \mapsto n] \quad \stackrel{\text{def}}{=} \begin{cases} (\emptyset \mid p \mapsto n) & \text{if } n > 0 \wedge p \in \text{Ptrs} \\ \emptyset & \text{if } n = 0 \wedge p = 0 \\ \text{undef} & \text{otherwise} \end{cases} \in \text{Shapes}$$

Note that if $n = 0$ and $[p \mapsto n]$ is defined, then $p = 0$.

$I_{\text{gc}} \in \mathbb{P}_{\text{fin}}(\text{Ptrs}) \times \text{Heaps} \rightarrow \text{Shapes}$ satisfying

(GC Axiom₁)

$$\forall R, h, \sigma = I_{\text{gc}}(R, h).$$

$$\overline{\text{dom}}(\sigma) \subseteq \text{dom}(h) \wedge \text{reach}(R, h, \sigma) \subseteq \text{dom}(\sigma)$$

(GC Axiom₂)

$$\forall R, h, \sigma = I_{\text{gc}}(R, h).$$

$$\forall R', h'. \overline{\text{dom}}(\sigma) \subseteq \text{dom}(h') \wedge \text{reach}(R', h', \sigma) \subseteq \text{dom}(\sigma) \wedge (\forall p \notin \overline{\text{dom}}(\sigma). h'(p) = h(p)) \implies$$

$$\exists \sigma' \subseteq \sigma. \sigma' = I_{\text{gc}}(R', h')$$

$$\forall s, h, \sigma, \mathbf{x}, n. \sigma = I_{\text{gc}}(\text{roots}(s), h) \wedge s(\mathbf{x}) = 2n + 1 \wedge n \geq 0 \implies$$

$$(\text{alloc } \mathbf{x}, s, h \rightsquigarrow -) \wedge$$

$$(\forall C', s', h'. \text{alloc } \mathbf{x}, s, h \rightsquigarrow C', s', h' \implies$$

$$\exists p, h'', \sigma''. C' = \text{skip} \wedge \sigma'' \uplus [p \mapsto n] = I_{\text{gc}}(\text{roots}(s'), h') \wedge s'(\mathbf{x}) = p \wedge h' = h'' \uplus [p \mapsto_n 0, \dots, 0] \wedge$$

$$(s, h, \sigma) \cong ((s' \mid \mathbf{x} \mapsto 2n + 1), h'', \sigma'')$$

2 Program Specifications

2.1 Logical Storage Model

$$\begin{aligned}
\text{Locs} &\stackrel{\text{def}}{=} \{ \ell \in \{ \text{loc}_1, \text{loc}_2, \dots \} \} \\
\text{LogPtrs} &\stackrel{\text{def}}{=} \{ \ell \hat{+} i \mid \ell \in \text{Locs} \wedge i \in \mathbb{Z} \} \\
\text{LogVals} &\stackrel{\text{def}}{=} \{ \mathbf{v} \in \text{LogPtrs} \uplus \text{Words} \} \\
\text{LStores} &\stackrel{\text{def}}{=} \{ \mathbf{s} \in \text{ProgVars} \rightarrow \text{LogVals} \} \\
\text{Span}(\mathbf{h}) &\stackrel{\text{def}}{=} \{ (\ell, i) \in \text{Locs} \times \mathbb{N} \mid i \in \text{dom}(\mathbf{h}(\ell)) \} \quad \text{for } \mathbf{h} \in \text{Locs} \rightarrow (\mathbb{N} \rightarrow_{\text{fin}} \text{LogVals}) \\
\text{LHeaps} &\stackrel{\text{def}}{=} \{ \mathbf{h} \in \text{Locs} \rightarrow \mathbb{N} \rightarrow_{\text{fin}} \text{LogVals} \mid \text{Span}(\mathbf{h}) \text{ is finite} \} \\
\text{Table} &\stackrel{\text{def}}{=} \{ \mathbf{T} \in \text{Locs} \rightarrow_{\text{fin}} \text{Ptrs} \times \mathbb{N}^+ \} \\
\text{phyv}_{\mathbf{T}}(\mathbf{v}) &\stackrel{\text{def}}{=} \begin{cases} w & \text{if } \mathbf{v} = w \in \text{Words} \\ p + i & \text{if } \mathbf{v} = \ell \hat{+} i \wedge \mathbf{T}(\ell) = (p, n) \\ \text{undef} & \text{otherwise} \end{cases} \\
\text{phyh}_{\mathbf{T}}(\mathbf{h}) &\stackrel{\text{def}}{=} \bigsqcup_{(p,n)=\mathbf{T}(\ell)} [p \mapsto_n \text{phyv}_{\mathbf{T}}(\mathbf{h}(\ell)(0)), \dots, \text{phyv}_{\mathbf{T}}(\mathbf{h}(\ell)(n-1))] \\
\text{shape}(\mathbf{T}) &\stackrel{\text{def}}{=} \bigsqcup_{(p,n)=\mathbf{T}(\ell)} [p \mapsto n] \\
\text{Safe}(\mathbf{L}) &\stackrel{\text{def}}{=} \{ \ell \hat{+} 0 \mid \ell \in \mathbf{L} \} \cup \text{NonPtrs} \quad \text{for } \mathbf{L} \subseteq \text{Locs} \\
\mathbf{s} \sim_{\mathbf{T}} s &\text{ iff } \forall \mathbf{x}. s(\mathbf{x}) = \text{phyv}_{\mathbf{T}}(\mathbf{s}(\mathbf{x})) \\
\mathbf{s} \approx_{\mathbf{T}} s &\text{ iff } \mathbf{s} \sim_{\mathbf{T}} s \wedge \forall \mathbf{x}. \mathbf{s}(\mathbf{x}) \in \text{Safe}(\text{dom}(\mathbf{T})) \\
\mathbf{h} : \mathbf{T} &\text{ iff } \forall \ell. \forall (p, n) = \mathbf{T}(\ell). \text{dom}(\mathbf{h}(\ell)) = \{ 0, \dots, n-1 \} \\
\mathbf{h} \sim_{\mathbf{T}} h &\text{ iff } \mathbf{h} : \mathbf{T} \wedge \text{phyh}_{\mathbf{T}}(\mathbf{h}) \subseteq h \\
\mathbf{h} :: \mathbf{T} &\text{ iff } \forall \ell. \forall (p, n) = \mathbf{T}(\ell). \forall i < n. \mathbf{h}(\ell)(i) \in \text{Safe}(\text{dom}(\mathbf{T})) \\
\mathbf{h} \approx_{\mathbf{T}} h &\text{ iff } \mathbf{h} \sim_{\mathbf{T}} h \wedge \mathbf{h} :: \mathbf{T} \wedge \text{shape}(\mathbf{T}) \subseteq I_{\text{gc}}(\text{dom}(\text{shape}(\mathbf{T})), h) \\
\mathbf{h}_1 \# \mathbf{h}_2 &\stackrel{\text{def}}{=} \text{Span}(\mathbf{h}_1) \cap \text{Span}(\mathbf{h}_2) = \emptyset \\
\mathbf{h}_1 \uplus \mathbf{h}_2 &\stackrel{\text{def}}{=} \begin{cases} \lambda \ell. \mathbf{h}_1(\ell) \uplus \mathbf{h}_2(\ell) & \text{if } \mathbf{h}_1 \# \mathbf{h}_2 \\ \text{undef} & \text{otherwise} \end{cases}
\end{aligned}$$

2.2 Syntax

Logical Expressions

$$\begin{aligned} \text{LogVars} &\stackrel{\text{def}}{=} \{u, v, \dots\} \\ \mathbf{E} \in \text{LExps} &::= v \\ &| \mathbf{x} \\ &| \mathbf{v} \\ &| \mathbf{E} \star \mathbf{E} \\ &| \text{not } \mathbf{E} \end{aligned}$$

where $v \in \text{LogVars}$, $\mathbf{x} \in \text{ProgVars}$, $\mathbf{v} \in \text{LogVals}$ and $\star \in \{+, -, \times, \div, <, =, \text{and}\}$

Note that $\text{Exps} \subseteq \text{LExps}$.

Assertions

$$\begin{aligned} P \in \text{Asserts} &::= \mathbf{E} \\ &| \mathbf{E} \leftrightarrow \mathbf{E} \quad | \quad P \star P \quad | \quad P \neg \star P \\ &| P \Rightarrow P \quad | \quad P \wedge P \quad | \quad P \vee P \\ &| \forall v. P \quad | \quad \exists v. P \end{aligned}$$

Assertions with safety

$$\begin{aligned} \mathbf{P} \in \text{AssertsL} &::= \text{safe}(\mathbf{E}) \\ &| \mathbf{E} \\ &| \mathbf{E} \leftrightarrow \mathbf{E} \quad | \quad \mathbf{P} \star \mathbf{P} \quad | \quad \mathbf{P} \neg \star \mathbf{P} \\ &| \mathbf{P} \Rightarrow \mathbf{P} \quad | \quad \mathbf{P} \wedge \mathbf{P} \quad | \quad \mathbf{P} \vee \mathbf{P} \\ &| \forall v. \mathbf{P} \quad | \quad \exists v. \mathbf{P} \end{aligned}$$

$$\begin{aligned} \text{false} &\stackrel{\text{def}}{=} 0; \quad \text{true} \stackrel{\text{def}}{=} 1; \quad \neg \mathbf{P} \stackrel{\text{def}}{=} \mathbf{P} \Rightarrow \text{false} \\ \text{defined}(\mathbf{E}) &\stackrel{\text{def}}{=} \mathbf{E} = \mathbf{E} \\ \text{word}(\mathbf{E}) &\stackrel{\text{def}}{=} \mathbf{E} = 0 \vee \mathbf{E} \\ \text{logptr}(\mathbf{E}) &\stackrel{\text{def}}{=} \text{defined}(\mathbf{E}) \wedge \neg(\text{word}(\mathbf{E})) \\ \text{nonptr}(\mathbf{E}) &\stackrel{\text{def}}{=} \mathbf{E} = 0 \vee \exists v. \mathbf{E} = 2 \times v + 1 \\ \text{offsafe}(\mathbf{E}) &\stackrel{\text{def}}{=} \text{word}(\mathbf{E}) \vee \exists i. \text{safe}(\mathbf{E} + i) \\ \text{p}(\{\mathbf{E}_1, \dots, \mathbf{E}_n\}) &\stackrel{\text{def}}{=} \text{p}(\mathbf{E}_1) \wedge \dots \wedge \text{p}(\mathbf{E}_n) \\ &\text{for } \text{p} = \text{safe}, \text{logptr}, \text{word}, \text{defined}, \text{nonptr}, \text{offsafe} \\ \mathbf{E} \leftrightarrow - &\stackrel{\text{def}}{=} \exists v. \mathbf{E} \leftrightarrow v \\ \mathbf{E} \leftrightarrow_n \mathbf{E}_0, \dots, \mathbf{E}_{n-1} &\stackrel{\text{def}}{=} \mathbf{E} + 4 \times 0 \leftrightarrow \mathbf{E}_0 \star \dots \star \mathbf{E} + 4(n-1) \leftrightarrow \mathbf{E}_{n-1} \end{aligned}$$

Note that $\text{Asserts} \subseteq \text{AssertsL}$.

Free variables

- $\text{FPV}(\mathbf{E}) \stackrel{\text{def}}{=} \text{the set of program variables appearing in the logical expression } \mathbf{E}$
 $\text{FLV}(\mathbf{E}) \stackrel{\text{def}}{=} \text{the set of free logical variables appearing in the assertion } \mathbf{E}$
 $\text{FPV}(\mathbf{P}) \stackrel{\text{def}}{=} \text{the set of program variables appearing in the assertion } \mathbf{P}$
 $\text{FLV}(\mathbf{P}) \stackrel{\text{def}}{=} \text{the set of free logical variables appearing in the assertion } \mathbf{P}$

Program Specifications

- $\{\mathbf{P}\} C \{\mathbf{Q}\} \quad :$ Inner-level partial correctness
 $[\mathbf{P}] C [\mathbf{Q}] \quad :$ Inner-level total correctness
 $\{\{P\}\} C \{\{Q\}\} \quad :$ Outer-level partial correctness
 $[\{P\}] C [\{Q\}] \quad :$ Outer-level total correctness

2.3 Semantics

$\llbracket \mathbf{E} \rrbracket \in \text{LStores} \rightarrow \text{LogVals}$

- $\llbracket v \rrbracket_s \quad ::= \text{undef}$
 $\llbracket \mathbf{x} \rrbracket_s \quad ::= \mathbf{s}(\mathbf{x})$
 $\llbracket \mathbf{v} \rrbracket_s \quad ::= \mathbf{v}$

$$\llbracket \mathbf{E}_1 \star \mathbf{E}_2 \rrbracket_s ::= \begin{cases} w_1 \star w_2 & \text{if } \llbracket \mathbf{E}_1 \rrbracket_s = w_1 \in \text{Words} \wedge \llbracket \mathbf{E}_2 \rrbracket_s = w_2 \in \text{Words} \\ \widehat{\ell} \hat{+} (i + w) & \text{if } \star = + \wedge \llbracket \mathbf{E}_k \rrbracket_s = \widehat{\ell} \hat{+} i \wedge \llbracket \mathbf{E}_{3-k} \rrbracket_s = w \text{ for } k = 1, 2 \\ \widehat{\ell} \hat{+} (i - w) & \text{if } \star = - \wedge \llbracket \mathbf{E}_1 \rrbracket_s = \widehat{\ell} \hat{+} i \wedge \llbracket \mathbf{E}_2 \rrbracket_s = w \\ i - j & \text{if } \star = - \wedge \llbracket \mathbf{E}_1 \rrbracket_s = \widehat{\ell} \hat{+} i \wedge \llbracket \mathbf{E}_2 \rrbracket_s = \widehat{\ell} \hat{+} j \\ i < j & \text{if } \star = < \wedge \llbracket \mathbf{E}_1 \rrbracket_s = \widehat{\ell} \hat{+} i \wedge \llbracket \mathbf{E}_2 \rrbracket_s = \widehat{\ell} \hat{+} j \\ i = j & \text{if } \star = = \wedge \llbracket \mathbf{E}_1 \rrbracket_s = \widehat{\ell} \hat{+} i \wedge \llbracket \mathbf{E}_2 \rrbracket_s = \widehat{\ell} \hat{+} j \\ \ell = \ell' & \text{if } \star = = \wedge \llbracket \mathbf{E}_1 \rrbracket_s = \widehat{\ell} \hat{+} 0 \wedge \llbracket \mathbf{E}_2 \rrbracket_s = \ell' \hat{+} 0 \\ 0 & \text{if } \star = = \wedge \llbracket \mathbf{E}_k \rrbracket_s = \widehat{\ell} \hat{+} 4i \wedge i \geq 0 \wedge \llbracket \mathbf{E}_{3-k} \rrbracket_s \in \text{NonPtrs} \text{ for } k = 1, 2 \\ \text{undef} & \text{otherwise} \end{cases}$$

where $\star \in \{+, -, \times, \div, <, =, \text{and}\}$, $w \div 0 = \text{undef}$,

$$w_1 < w_2 \stackrel{\text{def}}{=} 1 \text{ if } w_1 < w_2; 0 \text{ otherwise,}$$

$$w_1 = w_2 \stackrel{\text{def}}{=} 1 \text{ if } w_1 = w_2; 0 \text{ otherwise,}$$

$$w_1 \text{ and } w_2 \stackrel{\text{def}}{=} 1 \text{ if } w_1 \neq 0 \wedge w_2 \neq 0; 0 \text{ otherwise}$$

$$\llbracket \text{not } \mathbf{E} \rrbracket_s \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if } \llbracket \mathbf{E} \rrbracket_s = 0 \\ 0 & \text{if } \llbracket \mathbf{E} \rrbracket_s \in \text{NonPtrs} \setminus \{0\} \\ \text{undef} & \text{otherwise} \end{cases}$$

$\boxed{\mathbf{s}, \mathbf{h} \models_{\mathbf{L}} \mathbf{P}}$

$$\begin{aligned}
\mathbf{s}, \mathbf{h} \models_{\mathbf{L}} \text{safe}(\mathbf{E}) & \text{ iff } \llbracket \mathbf{E} \rrbracket_{\mathbf{s}} \in \text{Safe}(\mathbf{L}) \\
\mathbf{s}, \mathbf{h} \models_{\mathbf{L}} \mathbf{E} & \text{ iff } \llbracket \mathbf{E} \rrbracket_{\mathbf{s}} \in \text{Words} \setminus \{0\} \\
\mathbf{s}, \mathbf{h} \models_{\mathbf{L}} \mathbf{E}_1 \hookrightarrow \mathbf{E}_2 & \text{ iff } \exists \ell, i. \llbracket \mathbf{E}_1 \rrbracket_{\mathbf{s}} = \ell \widehat{\vdash} 4i \wedge \llbracket \mathbf{E}_2 \rrbracket_{\mathbf{s}} = \mathbf{h}(\ell)(i) \neq \text{undef} \\
\mathbf{s}, \mathbf{h} \models_{\mathbf{L}} \mathbf{P} * \mathbf{Q} & \text{ iff } \exists \mathbf{h}_1, \mathbf{h}_2. \mathbf{h} = \mathbf{h}_1 \uplus \mathbf{h}_2 \wedge \mathbf{s}, \mathbf{h}_1 \models_{\mathbf{L}} \mathbf{P} \wedge \mathbf{s}, \mathbf{h}_2 \models_{\mathbf{L}} \mathbf{Q} \\
\mathbf{s}, \mathbf{h} \models_{\mathbf{L}} \mathbf{P} \multimap \mathbf{Q} & \text{ iff } \forall \mathbf{h}'. \mathbf{h}' \# \mathbf{h} \wedge \mathbf{s}, \mathbf{h}' \models_{\mathbf{L}} \mathbf{P} \implies \mathbf{s}, \mathbf{h} \uplus \mathbf{h}' \models_{\mathbf{L}} \mathbf{Q} \\
\mathbf{s}, \mathbf{h} \models_{\mathbf{L}} \mathbf{P} \Rightarrow \mathbf{Q} & \text{ iff } \forall \mathbf{h}' \supseteq \mathbf{h}. \mathbf{s}, \mathbf{h}' \models_{\mathbf{L}} \mathbf{P} \implies \mathbf{s}, \mathbf{h}' \models_{\mathbf{L}} \mathbf{Q} \\
\mathbf{s}, \mathbf{h} \models_{\mathbf{L}} \mathbf{P} \wedge \mathbf{Q} & \text{ iff } \mathbf{s}, \mathbf{h} \models_{\mathbf{L}} \mathbf{P} \wedge \mathbf{s}, \mathbf{h} \models_{\mathbf{L}} \mathbf{Q} \\
\mathbf{s}, \mathbf{h} \models_{\mathbf{L}} \mathbf{P} \vee \mathbf{Q} & \text{ iff } \mathbf{s}, \mathbf{h} \models_{\mathbf{L}} \mathbf{P} \vee \mathbf{s}, \mathbf{h} \models_{\mathbf{L}} \mathbf{Q} \\
\mathbf{s}, \mathbf{h} \models_{\mathbf{L}} \forall v. \mathbf{P} & \text{ iff } \forall \mathbf{v} \in \text{LogVals}. \mathbf{s}, \mathbf{h} \models_{\mathbf{L}} \mathbf{P}[\mathbf{v}/v] \\
\mathbf{s}, \mathbf{h} \models_{\mathbf{L}} \exists v. \mathbf{P} & \text{ iff } \exists \mathbf{v} \in \text{LogVals}. \mathbf{s}, \mathbf{h} \models_{\mathbf{L}} \mathbf{P}[\mathbf{v}/v]
\end{aligned}$$

Note that

$$\begin{aligned}
\mathbf{s}, \mathbf{h} \models_{\mathbf{L}} \text{logptr}(\mathbf{E}) & \iff \llbracket \mathbf{E} \rrbracket_{\mathbf{s}} \in \text{LogPtrs} \\
\mathbf{s}, \mathbf{h} \models_{\mathbf{L}} \text{word}(\mathbf{E}) & \iff \llbracket \mathbf{E} \rrbracket_{\mathbf{s}} \in \text{Words}
\end{aligned}$$

$\boxed{\mathbf{s}, \mathbf{h} \models P}$

$$\mathbf{s}, \mathbf{h} \models P \text{ iff } \mathbf{s}, \mathbf{h} \models_{\emptyset} P$$

Notation

$$\begin{aligned}
\mathbf{P}[\rho] & \stackrel{\text{def}}{=} \mathbf{P}[\rho(v_1)/v_1] \dots [\rho(v_n)/v_n] \text{ where } \text{dom}(\rho) = \{v_1, \dots, v_n\} \text{ for } \rho \in \text{LogVars} \rightarrow_{\text{fin}} \text{LogVals} \\
\text{Env}(V) & \stackrel{\text{def}}{=} \{\rho \in \text{LogVars} \rightarrow_{\text{fin}} \text{LogVals} \mid \text{dom}(\rho) \supseteq V\} \text{ for } V \subseteq_{\text{fin}} \text{LogVars} \\
\rho|_V(\mathbf{x}) & \stackrel{\text{def}}{=} \begin{cases} \rho(\mathbf{x}) & \text{if } \mathbf{x} \in \text{dom}(\rho) \\ 0 & \text{else if } \mathbf{x} \in V \\ \text{undef} & \text{otherwise} \end{cases}
\end{aligned}$$

$\boxed{\mathbf{P} \models \mathbf{Q}}$

$$\begin{aligned}
\mathbf{P} \models \mathbf{Q} & \text{ iff } \forall \rho \in \text{Env}(\text{FLV}(\mathbf{P}, \mathbf{Q})), \mathbf{s}, \mathbf{h}, \mathbf{h}_F, \mathbf{T}, s, h. \\
& \mathbf{s} \sim_{\mathbf{T}} s \wedge \mathbf{h} \uplus \mathbf{h}_F \approx_{\mathbf{T}} h \wedge \mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} \mathbf{P}[\rho] \implies \mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} \mathbf{Q}[\rho]
\end{aligned}$$

$\boxed{\{\mathbf{P}\} C \{\mathbf{Q}\}}$

$$\begin{aligned}
\{\mathbf{P}\} C \{\mathbf{Q}\} & \text{ iff } \forall \rho \in \text{Env}(\text{FLV}(\mathbf{P}, \mathbf{Q})), \mathbf{s}, \mathbf{h}, \mathbf{h}_F, \mathbf{T}, s, h, C', s', h'. \\
& \mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} \mathbf{P}[\rho] \wedge \mathbf{s} \sim_{\mathbf{T}} s \wedge \mathbf{h} \uplus \mathbf{h}_F \approx_{\mathbf{T}} h \wedge C, s, h \rightsquigarrow^* C', s', h' \implies \\
& (C', s', h' \rightsquigarrow -) \vee \\
& (\exists s', h'. C' = \text{skip} \wedge s', h' \models_{\text{dom}(\mathbf{T})} \mathbf{Q}[\rho] \wedge \\
& (\forall \mathbf{x} \notin \text{Mod}(C). s'(\mathbf{x}) = \mathbf{s}(\mathbf{x})) \wedge s' \sim_{\mathbf{T}} s' \wedge h' \uplus \mathbf{h}_F \approx_{\mathbf{T}} h')
\end{aligned}$$

$\boxed{[P] C [Q]}$

$$\begin{aligned}
[P] C [Q] \text{ iff } & \{P\} C \{Q\} \wedge \\
& \forall \rho \in \text{Env}(\text{FLV}(P, Q)), \mathbf{s}, \mathbf{h}, \mathbf{h}_F, \mathbf{T}, s, h. \\
& \mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} P[\rho] \wedge \mathbf{s} \sim_{\mathbf{T}} s \wedge \mathbf{h} \uplus \mathbf{h}_F \approx_{\mathbf{T}} h \implies \neg(C, s, h \text{ diverges})
\end{aligned}$$

$\boxed{\{\{P\}\} C \{\{Q\}\}}$

$$\begin{aligned}
\{\{P\}\} C \{\{Q\}\} \text{ iff } & \forall \rho \in \text{Env}(\text{FLV}(P, Q)), \mathbf{s}, \mathbf{h}, \mathbf{h}_F, \mathbf{T}, s, h, C', s', h'. \\
& \mathbf{s}, \mathbf{h} \models P[\rho] \wedge \mathbf{s} \approx_{\mathbf{T}} s \wedge \mathbf{h} \uplus \mathbf{h}_F \approx_{\mathbf{T}} h \wedge C, s, h \rightsquigarrow^* C', s', h' \implies \\
& (C', s', h' \rightsquigarrow -) \vee \\
& (\exists \mathbf{s}', \mathbf{h}', \mathbf{T}'. C' = \text{skip} \wedge \mathbf{s}', \mathbf{h}' \models Q[\rho] \wedge \\
& (\forall \mathbf{x} \notin \text{Mod}(C). \mathbf{s}'(\mathbf{x}) = \mathbf{s}(\mathbf{x})) \wedge \mathbf{s}' \approx_{\mathbf{T}'} s' \wedge \mathbf{h}' \uplus \mathbf{h}_F \approx_{\mathbf{T}'} h')
\end{aligned}$$

$\boxed{[[P]] C [[Q]]}$

$$\begin{aligned}
[[P]] C [[Q]] \text{ iff } & \{\{P\}\} C \{\{Q\}\} \wedge \\
& \forall \rho \in \text{Env}(\text{FLV}(P, Q)), \mathbf{s}, \mathbf{h}, \mathbf{h}_F, \mathbf{T}, s, h. \\
& \mathbf{s}, \mathbf{h} \models P[\rho] \wedge \mathbf{s} \approx_{\mathbf{T}} s \wedge \mathbf{h} \uplus \mathbf{h}_F \approx_{\mathbf{T}} h \implies \neg(C, s, h \text{ diverges})
\end{aligned}$$

3 Program Logic

3.1 Inner-level rules

$$\begin{array}{c}
\frac{}{[\text{true}] \text{ skip } [\text{true}]} \quad (\text{Skip}) \\
\\
\frac{}{[x = v \wedge \text{defined}(E)] \ x := E \ [x = E[v/x]]} \quad (\text{Assign}) \\
\\
\frac{}{[x = u \wedge E \hookrightarrow v] \ x := [E] \ [x = v \wedge E[u/x] \hookrightarrow v]} \quad (\text{Read}) \\
\\
\frac{}{[E \hookrightarrow - \wedge \text{safe}(E')] \ [E] := E' \ [E \hookrightarrow E']} \quad (\text{Write}) \\
\\
\frac{\frac{\{P \wedge E\} C_1 \ \{Q\} \quad \{P \wedge \text{not } E\} C_2 \ \{Q\}}{\{P \wedge \text{word}(E)\} \text{ if } E \text{ then } C_1 \text{ else } C_2 \text{ fi } \ \{Q\}} \quad \frac{[P \wedge E] C_1 \ [Q] \quad [P \wedge \text{not } E] C_2 \ [Q]}{[P \wedge \text{word}(E)] \text{ if } E \text{ then } C_1 \text{ else } C_2 \text{ fi } \ [Q]} \quad (\text{If})}{} \\
\\
\frac{\frac{\{P \wedge E\} C \ \{P \wedge \text{word}(E)\}}{\{P \wedge \text{word}(E)\} \text{ while } E \text{ do } C \ \text{od } \ \{P \wedge \text{not } E\}} \quad (\text{While})}{} \\
\\
\frac{\frac{[P \wedge E \wedge 0 < E' = v] C \ [P \wedge \text{word}(E) \wedge 0 < E' < v] \quad v \notin \text{FLV}(P, E')}{[P \wedge \text{word}(E) \wedge 0 < E'] \text{ while } E \text{ do } C \ \text{od } \ [P \wedge \text{not } E]} \quad (\text{WhileT})}{} \\
\\
\frac{\frac{\{P\} C_1 \ \{Q\} \quad \{Q\} C_2 \ \{R\}}{\{P\} C_1; C_2 \ \{R\}} \quad \frac{[P] C_1 \ [Q] \quad [Q] C_2 \ [R]}{[P] C_1; C_2 \ [R]}} \quad (\text{Seq})}{} \\
\\
\frac{\frac{\{P\} C \ \{Q\} \quad \text{FPV}(\mathbf{R}) \cap \text{Mod}(C) = \emptyset}{\{P * \mathbf{R}\} C \ \{Q * \mathbf{R}\}} \quad \frac{[P] C \ [Q] \quad \text{FPV}(\mathbf{R}) \cap \text{Mod}(C) = \emptyset}{[P * \mathbf{R}] C \ [Q * \mathbf{R}]}} \quad (\text{Frame})}{} \\
\\
\frac{P \models P' \quad \frac{\{P'\} C \ \{Q'\}}{\{P\} C \ \{Q\}} \quad Q' \models Q}{\frac{P \models P' \quad [P'] C \ [Q'] \quad Q' \models Q}{[P] C \ [Q]}} \quad (\text{Conseq})}{} \\
\\
\frac{\frac{\{P\} C \ \{Q\}}{\{\exists v. P\} C \ \{\exists v. Q\}} \quad \frac{[P] C \ [Q]}{[\exists v. P] C \ [\exists v. Q]}} \quad (\text{Ex})}{} \\
\\
\frac{\frac{\forall v \in \text{LogVals. } \{P[v/v]\} C \ \{Q[v/v]\}}{\{P\} C \ \{Q\}} \quad \frac{\forall v \in \text{LogVals. } [P[v/v]] C \ [Q[v/v]]}{[P] C \ [Q]}} \quad (\text{Gen})}{} \\
\\
\frac{[P] C \ [Q]}{\{P\} C \ \{Q\}} \quad (\text{Total})
\end{array}$$

3.2 Outer-level rules

$$\frac{n \geq 0}{[[x = 2n + 1]] \text{ alloc } x \ [[x \hookrightarrow_n 0, \dots, 0]]} \quad (\text{Alloc})$$

$$\frac{V \subseteq_{\text{fin}} \text{ProgVars} \quad \{P \wedge \text{safe}(V)\} C \{Q \wedge \text{safe}(\text{Mod}(C))\}}{\{\{P\}\} C \{\{Q\}\}} \quad (\text{Incl})$$

$$\frac{V \subseteq_{\text{fin}} \text{ProgVars} \quad [P \wedge \text{safe}(V)] C [Q \wedge \text{safe}(\text{Mod}(C))]}{[[P]] C [[Q]]}$$

$$\frac{\{\{P \wedge E\}\} C_1 \{\{Q\}\} \quad \{\{P \wedge \text{not } E\}\} C_2 \{\{Q\}\}}{\{\{P \wedge \text{word}(E)\}\} \text{ if } E \text{ then } C_1 \text{ else } C_2 \text{ fi } \{\{Q\}\}} \quad \frac{[[P \wedge E]] C_1 [[Q]] \quad [[P \wedge \text{not } E]] C_2 [[Q]]}{[[P \wedge \text{word}(E)]] \text{ if } E \text{ then } C_1 \text{ else } C_2 \text{ fi } [[Q]]} \quad (\text{If})$$

$$\frac{\{\{P \wedge E\}\} C \{\{P \wedge \text{word}(E)\}\}}{\{\{P \wedge \text{word}(E)\}\} \text{ while } E \text{ do } C \text{ od } \{\{P \wedge \text{not } E\}\}} \quad (\text{While})$$

$$\frac{[[P \wedge E \wedge 0 < \mathbf{E}' = v]] C [[P \wedge \text{word}(E) \wedge 0 < \mathbf{E}' < v]] \quad v \notin \text{FLV}(P, \mathbf{E}')}{[[P \wedge \text{word}(E) \wedge 0 < \mathbf{E}']] \text{ while } E \text{ do } C \text{ od } [[P \wedge \text{not } E]]} \quad (\text{WhileT})$$

$$\frac{\frac{\{\{P\}\} C_1 \{\{Q\}\} \quad \{\{Q\}\} C_2 \{\{R\}\}}{\{\{P\}\} C_1; C_2 \{\{R\}\}} \quad \frac{[[P]] C_1 [[Q]] \quad [[Q]] C_2 [[R]]}{[[P]] C_1; C_2 [[R]]}}{\{\{P\}\} C_1; C_2 \{\{R\}\}} \quad (\text{Seq})$$

$$\frac{\frac{\{\{P\}\} C \{\{Q\}\} \quad \text{FPV}(R) \cap \text{Mod}(C) = \emptyset}{\{\{P * R\}\} C \{\{Q * R\}\}} \quad \frac{[[P]] C [[Q]] \quad \text{FPV}(R) \cap \text{Mod}(C) = \emptyset}{[[P * R]] C [[Q * R]]}}{\{\{P * R\}\} C \{\{Q * R\}\}} \quad (\text{Frame})$$

$$\frac{P \models P' \quad \frac{\{\{P'\}\} C \{\{Q'\}\} \quad Q' \models Q}{\{\{P'\}\} C \{\{Q'\}\}} \quad \frac{P \models P' \quad \frac{[[P']] C [[Q']] \quad Q' \models Q}{[[P']] C [[Q']]]}{[[P]] C [[Q]]} \quad (\text{Conseq})$$

$$\frac{\frac{\{\{P\}\} C \{\{Q\}\}}{\{\{\exists v. P\}\} C \{\{\exists v. Q\}\}} \quad \frac{[[P]] C [[Q]]}{[[\exists v. P]] C [[\exists v. Q]]}}{\{\{\exists v. P\}\} C \{\{\exists v. Q\}\}} \quad (\text{Ex})$$

$$\frac{\forall \mathbf{v} \in \text{LogVals. } \{\{P[\mathbf{v}/v]\}\} C \{\{Q[\mathbf{v}/v]\}\}}{\{\{P\}\} C \{\{Q\}\}} \quad \frac{\forall \mathbf{v} \in \text{LogVals. } [[P[\mathbf{v}/v]]] C [[Q[\mathbf{v}/v]]]}{[[P]] C [[Q]]} \quad (\text{Gen})$$

$$\frac{[[P]] C [[Q]]}{\{\{P\}\} C \{\{Q\}\}} \quad (\text{Total})$$

3.3 Assertion entailments

$$\begin{array}{l}
\text{nonptr}(\mathbf{E}) \models \text{safe}(\mathbf{E}) \quad (\text{NPtrSafe}) \\
\mathbf{E} \models \text{word}(\mathbf{E}) \quad (\text{BoolWord}) \\
\mathbf{E} \hookrightarrow \mathbf{E}' \models \mathbf{E} \neq 0 \quad (\text{PointstoNZero}) \\
\text{defined}(E) \models \text{offsafe}(E) \quad (\text{ExpSafe}) \\
\mathbf{E} \hookrightarrow \mathbf{E}' \wedge \text{offsafe}(\mathbf{E}) \models \text{safe}(\mathbf{E}') \quad (\text{HeapSafe}) \\
E \hookrightarrow \mathbf{E}' \models \text{safe}(\mathbf{E}') \quad (\text{ExpHeapSafe}) \\
\text{safe}(\mathbf{E}, \mathbf{E}') \models \text{defined}(\mathbf{E} = \mathbf{E}') \quad (\text{SafeEq})
\end{array}$$

3.4 Derived rules

$$\begin{array}{l}
\frac{\{\mathbf{P}\} C \{\mathbf{Q}\} \quad v \notin \text{FLV}(\mathbf{Q})}{\{\exists v. \mathbf{P}\} C \{\mathbf{Q}\}} \quad \frac{[\mathbf{P}] C [\mathbf{Q}] \quad v \notin \text{FLV}(\mathbf{Q})}{[\exists v. \mathbf{P}] C [\mathbf{Q}]} \quad (\text{Ex}') \\
\frac{\{\{P\}\} C \{\{Q\}\} \quad v \notin \text{FLV}(Q)}{\{\{\exists v. P\}\} C \{\{Q\}\}} \quad \frac{[[P]] C [[Q]] \quad v \notin \text{FLV}(Q)}{[[\exists v. P]] C [[Q]]} \quad (\text{Ex}') \\
\frac{\{\mathbf{P}_1\} C \{\mathbf{Q}\} \quad \{\mathbf{P}_2\} C \{\mathbf{Q}\}}{\{\mathbf{P}_1 \vee \mathbf{P}_2\} C \{\mathbf{Q}\}} \quad \frac{[\mathbf{P}_1] C [\mathbf{Q}] \quad [\mathbf{P}_2] C [\mathbf{Q}]}{[\mathbf{P}_1 \vee \mathbf{P}_2] C [\mathbf{Q}]} \quad (\text{Disj}) \\
\frac{\{\{P_1\}\} C \{\{Q\}\} \quad \{\{P_2\}\} C \{\{Q\}\}}{\{\{P_1 \vee P_2\}\} C \{\{Q\}\}} \quad \frac{[[P_1]] C [[Q]] \quad [[P_2]] C [[Q]]}{[[P_1 \vee P_2]] C [[Q]]} \quad (\text{Disj}) \\
\frac{\{\mathbf{P}\} C \{\mathbf{Q}\} \quad \text{FPV}(\mathbf{E}) \cap \text{Mod}(C) = \emptyset}{\{\mathbf{P}[\mathbf{E}/v] \wedge \text{defined}(\mathbf{E})\} C \{\mathbf{Q}[\mathbf{E}/v]\}} \quad \frac{[\mathbf{P}] C [\mathbf{Q}] \quad \text{FPV}(\mathbf{E}) \cap \text{Mod}(C) = \emptyset}{[\mathbf{P}[\mathbf{E}/v] \wedge \text{defined}(\mathbf{E})] C [\mathbf{Q}[\mathbf{E}/v]]} \quad (\text{Inst}) \\
\frac{\{\{P\}\} C \{\{Q\}\} \quad \text{FPV}(\mathbf{E}) \cap \text{Mod}(C) = \emptyset}{\{\{P[\mathbf{E}/v] \wedge \text{defined}(\mathbf{E})\}\} C \{\{Q[\mathbf{E}/v]\}\}} \quad \frac{[[P]] C [[Q]] \quad \text{FPV}(\mathbf{E}) \cap \text{Mod}(C) = \emptyset}{[[P[\mathbf{E}/v] \wedge \text{defined}(\mathbf{E})]] C [[Q[\mathbf{E}/v]]]} \quad (\text{Inst}) \\
\frac{}{[\mathbf{P}[E/x] \wedge \text{defined}(E)] \quad x := E \quad [\mathbf{P}]} \quad (\text{Assign}') \\
\frac{x \notin \text{FPV}(E) \cup \text{FPV}(\mathbf{E}')}{[E \hookrightarrow \mathbf{E}'] \quad x := [E] \quad [x = \mathbf{E}' \wedge E \hookrightarrow \mathbf{E}']} \quad (\text{Read}') \\
\frac{x \notin \text{FPV}(\mathbf{E}') \cup \text{FPV}(\mathbf{E}'')}{[x = \mathbf{E}' \wedge E \hookrightarrow \mathbf{E}''] \quad x := [E] \quad [x = \mathbf{E}'' \wedge E[\mathbf{E}'/x] \hookrightarrow \mathbf{E}'']} \quad (\text{Read}'')
\end{array}$$

$$\frac{}{\overline{[[\mathbf{P}[y/x]] \text{ x} := y \text{ } [[\mathbf{P}]]}} \quad (\text{ASSIGN})$$

$$\frac{}{\overline{[[\mathbf{P}[E/x] \wedge \text{nonptr}(E)] \text{ x} := E \text{ } [[\mathbf{P}]]}} \quad (\text{ASSIGN}')$$

$$\frac{x \notin \text{FPV}(E) \cup \text{FPV}(\mathbf{E}')}{\overline{[[E \hookrightarrow \mathbf{E}']] \text{ x} := [E] \text{ } [[x = \mathbf{E}' \wedge E \hookrightarrow \mathbf{E}']]}} \quad (\text{READ})$$

$$\frac{x \notin \text{FPV}(\mathbf{E}') \cup \text{FPV}(\mathbf{E}'')}{\overline{[[x = \mathbf{E}' \wedge E \hookrightarrow \mathbf{E}'']] \text{ x} := [E] \text{ } [[x = \mathbf{E}'' \wedge E[\mathbf{E}'/x] \hookrightarrow \mathbf{E}'']]}} \quad (\text{READ}')$$

$$\frac{}{\overline{[[E \hookrightarrow -]] [E] := x \text{ } [[E \hookrightarrow x]]}} \quad (\text{WRITE})$$

$$\frac{}{\overline{[[E \hookrightarrow - \wedge \text{nonptr}(E')]] [E] := E' \text{ } [[E \hookrightarrow E']]}} \quad (\text{WRITE}')$$

$$\frac{n \geq 0}{\overline{[[E = 2n + 1]] \text{ x} := \text{ALLOC}(E) \text{ } [[x \hookrightarrow_n 0, \dots, 0]]}} \quad (\text{ALLOC})$$

3.5 Problematic rules

$$\frac{\frac{\{\mathbf{P}\} C \{\mathbf{Q}_1\} \quad \{\mathbf{P}\} C \{\mathbf{Q}_2\}}{\{\mathbf{P}\} C \{\mathbf{Q}_1 \wedge \mathbf{Q}_2\}} \quad \frac{[\mathbf{P}] C [\mathbf{Q}_1] \quad [\mathbf{P}] C [\mathbf{Q}_2]}{[\mathbf{P}] C [\mathbf{Q}_1 \wedge \mathbf{Q}_2]}}{\text{(Conj)}} \quad (\text{Conj})$$

$$\frac{\frac{\{\{P\}\} C \{\{Q_1\}\} \quad \{\{P\}\} C \{\{Q_2\}\}}{\{\{P\}\} C \{\{Q_1 \wedge Q_2\}\}} \quad \frac{[[P]] C [[Q_1]] \quad [[P]] C [[Q_2]]}{[[P]] C [[Q_1 \wedge Q_2]]}}{\text{(Conj)}} \quad (\text{Conj})$$

$$\frac{\frac{\{\mathbf{P}\} C \{\mathbf{Q}\}}{\{\forall v. \mathbf{P}\} C \{\forall v. \mathbf{Q}\}} \quad \frac{[\mathbf{P}] C [\mathbf{Q}]}{[\forall v. \mathbf{P}] C [\forall v. \mathbf{Q}]}}{\text{(All)}} \quad (\text{All})$$

$$\frac{\frac{\{\{P\}\} C \{\{Q\}\}}{\{\{\forall v. P\}\} C \{\{\forall v. Q\}\}} \quad \frac{[[P]] C [[Q]]}{[[\forall v. P]] C [[\forall v. Q]]}}{\text{(All)}} \quad (\text{All})$$

Counter example. According to the semantics of $\{-\} - \{-\}$, the following hold:

$$\begin{aligned} & \{x = 0 \wedge y \hookrightarrow 0\} \text{ x} := x \text{ } \{x = 0\} \\ & \{x = 0 \wedge y \hookrightarrow 0\} \text{ x} := x \text{ } \{\text{logptr}(x)\} \end{aligned}$$

However, the following conjunction does NOT hold:

$$\{x = 0 \wedge y \hookrightarrow 0\} \text{ x} := x \text{ } \{x = 0 \wedge \text{logptr}(x)\}$$

4 Examples

4.1 Array Assignment

$$\begin{aligned} & \{\{y + 8 \leftrightarrow -\}\} \\ & \{y + 8 \leftrightarrow - \wedge \underline{\text{safe}(y)}\} && \text{(Incl)} \\ & \mathbf{y := y + 8;} \\ & \{y \leftrightarrow - \wedge \underline{\text{safe}(y - 8)}\} && \text{(Assign')} \\ & \{y \leftrightarrow - \wedge \underline{\text{safe}(y - 8, 0)}\} \\ & \mathbf{[y] := 0;} \\ & \{y \leftrightarrow 0 \wedge \underline{\text{safe}(y - 8)}\} && \text{(Write)} \\ & \mathbf{y := y - 8;} \\ & \{y + 8 \leftrightarrow 0 \wedge \underline{\text{safe}(y)}\} && \text{(Assign')} \\ & \{\{y + 8 \leftrightarrow 0\}\} && \text{(Incl)} \end{aligned}$$

4.2 Word Swap

$$\begin{aligned} & \{\{x \leftrightarrow_2 u, v\}\} \\ & \mathbf{t := ALLOC(ENC(0))} \\ & \{\{x \leftrightarrow_2 u, v * t \leftrightarrow_0 \cdot\}\} && \text{(ALLOC)} \\ & \{\{x \leftrightarrow_2 u, v\}\} \\ & \mathbf{t := [x];} \\ & \{\{x \leftrightarrow_2 u, v \wedge t = u\}\} && \text{(READ)} \\ & \mathbf{r := [x + 4];} \\ & \{\{x \leftrightarrow_2 u, v \wedge t = u \wedge r = v\}\} && \text{(READ)} \\ & \mathbf{[x] := r;} \\ & \{\{x \leftrightarrow_2 r, v \wedge t = u \wedge r = v\}\} && \text{(WRITE)} \\ & \mathbf{[x + 4] := t;} \\ & \{\{x \leftrightarrow_2 r, t \wedge t = u \wedge r = v\}\} && \text{(WRITE)} \\ & \{\{x \leftrightarrow_2 v, u\}\} \end{aligned}$$

4.3 Linking of Assignment and Swap

From Sections 4.1 and 4.2, we have the following results.

$$\begin{aligned}
\text{Assign} &\stackrel{\text{def}}{=} y := y + 8; [y] := 0; y := y - 8 \\
\text{Swap} &\stackrel{\text{def}}{=} t := 1; \text{alloc } t; t := [x]; r := [x + 4]; [x] := r; [x + 4] := t \\
&\{\{y + 8 \leftrightarrow -\}\} \text{Assign} \{\{y + 8 \leftrightarrow 0\}\} \\
&\{\{x \leftrightarrow_2 u, v\}\} \text{Swap} \{\{x \leftrightarrow_2 v, u\}\}
\end{aligned}$$

From these, we can reason about the linked program as follows.

$$\begin{aligned}
&\frac{\{\{y + 8 \leftrightarrow -\}\} \text{Assign} \{\{y + 8 \leftrightarrow 0\}\} \quad \text{FPV}(x \leftrightarrow_2 u, v) \cap \text{Mod}(\text{Assign}) = \emptyset}{\{\{x \leftrightarrow_2 u, v * y + 8 \leftrightarrow -\}\} \text{Assign} \{\{x \leftrightarrow_2 u, v * y + 8 \leftrightarrow 0\}\}} \text{ (Frame)} \\
&\frac{\{\{x \leftrightarrow_2 u, v\}\} \text{Swap} \{\{x \leftrightarrow_2 v, u\}\} \quad \text{FPV}(y + 8 \leftrightarrow 0) \cap \text{Mod}(\text{Swap}) = \emptyset}{\{\{x \leftrightarrow_2 u, v * y + 8 \leftrightarrow 0\}\} \text{Swap} \{\{x \leftrightarrow_2 v, u * y + 8 \leftrightarrow 0\}\}} \text{ (Frame)} \\
&\frac{\{\{x \leftrightarrow_2 u, v * y + 8 \leftrightarrow -\}\} \text{Assign} \{\{x \leftrightarrow_2 u, v * y + 8 \leftrightarrow 0\}\} \\
&\quad \{\{x \leftrightarrow_2 u, v * y + 8 \leftrightarrow 0\}\} \text{Swap} \{\{x \leftrightarrow_2 v, u * y + 8 \leftrightarrow 0\}\}}{\{\{x \leftrightarrow_2 u, v * y + 8 \leftrightarrow -\}\} \text{Assign}; \text{Swap} \{\{x \leftrightarrow_2 v, u * y + 8 \leftrightarrow 0\}\}} \text{ (Seq)}
\end{aligned}$$

4.4 Simple Addition

$$\begin{aligned}
&\{\{x = 2 \times n + 1 \wedge y = 2 \times m + 1\}\} \\
&\{x = 2 \times n + 1 \wedge y = 2 \times m + 1\} \tag{Incl} \\
&\{x + y = 2 \times (n + m) + 2 \wedge x = 2 \times n + 1 \wedge y = 2 \times m + 1\} \\
&\mathbf{z := x + y;} \\
&\{z = 2 \times (n + m) + 2 \wedge x = 2 \times n + 1 \wedge y = 2 \times m + 1\} \tag{Assign'} \\
&\{z - 1 = 2 \times (n + m) + 1 \wedge x = 2 \times n + 1 \wedge y = 2 \times m + 1\} \\
&\mathbf{z := z - 1;} \\
&\{z = 2 \times (n + m) + 1 \wedge x = 2 \times n + 1 \wedge y = 2 \times m + 1\} \tag{Assign'} \\
&\{z = 2 \times (n + m) + 1 \wedge x = 2 \times n + 1 \wedge y = 2 \times m + 1 \wedge \text{safe}(z)\} \\
&\{\{z = 2 \times (n + m) + 1 \wedge x = 2 \times n + 1 \wedge y = 2 \times m + 1\}\} \tag{Incl}
\end{aligned}$$

4.5 Integer Arithmetic

Simple version

$$\{\{x = 2 \times n + 1 \wedge y = 2 \times m + 1\}\}$$

$$\begin{aligned}
& \{\{2 \times ((x \div 2 + y \div 2) \times (x \div 2 + y \div 2) \times (x \div 2 + y \div 2) \times (x \div 2 + y \div 2)) + 1 = \\
& \quad 2 \times (m + n) \times (m + n) \times (m + n) \times (m + n) + 1\}\} \\
& \mathbf{x} := 2 \times ((x \div 2 + y \div 2) \times (x \div 2 + y \div 2) \times (x \div 2 + y \div 2) \times (x \div 2 + y \div 2)) + 1 \\
& \{\{\mathbf{x} = 2 \times (m + n) \times (m + n) \times (m + n) \times (m + n) + 1\}\} \quad (\text{ASSIGN}')
\end{aligned}$$

Optimized version

$$\begin{aligned}
& \{\{\mathbf{x} = 2 \times n + 1 \wedge \mathbf{y} = 2 \times m + 1\}\} \\
& \{\mathbf{x} = 2 \times n + 1 \wedge \mathbf{y} = 2 \times m + 1\} \quad (\text{Incl}) \\
& \{\mathbf{x} + \mathbf{y} = 2 \times (n + m) + 2\} \\
& \mathbf{x} := \mathbf{x} + \mathbf{y}; \\
& \{\mathbf{x} = 2 \times (n + m) + 2\} \quad (\text{Assign}') \\
& \{\mathbf{x} \div 2 - 1 = n + m \wedge \text{word}(n, m)\} \\
& \mathbf{x} := \mathbf{x} \div 2 - 1; \\
& \{\mathbf{x} = n + m \wedge \text{word}(n, m)\} \quad (\text{Assign}') \\
& \{\mathbf{x} \times \mathbf{x} = (n + m) \times (n + m)\} \\
& \mathbf{x} := \mathbf{x} \times \mathbf{x}; \\
& \{\mathbf{x} = (n + m) \times (n + m)\} \quad (\text{Assign}') \\
& \{\mathbf{x} \times \mathbf{x} = (n + m) \times (n + m) \times (n + m) \times (n + m)\} \\
& \mathbf{x} := \mathbf{x} \times \mathbf{x}; \\
& \{\mathbf{x} = (n + m) \times (n + m) \times (n + m) \times (n + m)\} \quad (\text{Assign}') \\
& \{2 \times \mathbf{x} + 1 = 2 \times (n + m) \times (n + m) \times (n + m) \times (n + m) + 1\} \\
& \mathbf{x} := 2 \times \mathbf{x} + 1 \\
& \{\mathbf{x} = 2 \times (n + m) \times (n + m) \times (n + m) \times (n + m) + 1\} \quad (\text{Assign}') \\
& \{\mathbf{x} = 2 \times (m + n) \times (m + n) \times (m + n) \times (m + n) + 1 \wedge \underline{\text{safe}(\mathbf{x})}\} \\
& \{\{\mathbf{x} = 2 \times (m + n) \times (m + n) \times (m + n) \times (m + n) + 1\}\} \quad (\text{Incl})
\end{aligned}$$

4.6 List Reversal

$$\begin{aligned}
\epsilon^\dagger & \stackrel{\text{def}}{=} \epsilon \\
(v \cdot \alpha)^\dagger & \stackrel{\text{def}}{=} \alpha^\dagger \cdot v \\
\text{list } \epsilon \mathbf{E} & \stackrel{\text{def}}{=} \mathbf{E} = 0 \\
\text{list } (v \cdot \alpha) \mathbf{E} & \stackrel{\text{def}}{=} \exists z. (\mathbf{E} \hookrightarrow_2 v, z) * \text{list } \alpha z
\end{aligned}$$

$$\{\{\text{list } \alpha_0 \mathbf{x}\}\}$$

$\{\{\text{list } \alpha_0 \mathbf{x} * 0 = 0\} \wedge \underline{\text{defined}}(0)\}$
y := 0;
 $\{\{\text{list } \alpha_0 \mathbf{x} * \mathbf{y} = 0\}\}$ (ASSIGN')
 $\{\{\text{list } \alpha_0 \mathbf{x} * \text{list } \epsilon \mathbf{y}\}\}$
 $\{\{\exists \alpha, \beta. (\text{list } \alpha \mathbf{x} * \text{list } \beta \mathbf{y}) \wedge \alpha_0^\dagger = \alpha^\dagger \cdot \beta \wedge \underline{\text{word}}(\mathbf{x} \neq 0)\}\}$
while x ≠ 0 do
 $\{\{\exists \alpha, \beta. (\text{list } \alpha \mathbf{x} * \text{list } \beta \mathbf{y}) \wedge \alpha_0^\dagger = \alpha^\dagger \cdot \beta \wedge \mathbf{x} \neq 0\}\}$ (While)
 $\{\{\exists v, \alpha, \beta. (\text{list } (v \cdot \alpha) \mathbf{x} * \text{list } \beta \mathbf{y}) \wedge \alpha_0^\dagger = (v \cdot \alpha)^\dagger \cdot \beta\}\}$
 $\{\{\exists v, \alpha, \beta, z. (\mathbf{x} \hookrightarrow_2 v, z * \text{list } \alpha z * \text{list } \beta \mathbf{y}) \wedge \alpha_0^\dagger = (v \cdot \alpha)^\dagger \cdot \beta\}\}$
 $\{\{\exists v, \alpha, \beta. (\mathbf{x} \hookrightarrow_2 v, z * \text{list } \alpha z * \text{list } \beta \mathbf{y}) \wedge \alpha_0^\dagger = (v \cdot \alpha)^\dagger \cdot \beta\}\}$ (Ex')
z := [x + 4];
 $\{\{\exists v, \alpha, \beta. \mathbf{z} = z \wedge (\mathbf{x} \hookrightarrow_2 v, z * \text{list } \alpha z * \text{list } \beta \mathbf{y}) \wedge \alpha_0^\dagger = (v \cdot \alpha)^\dagger \cdot \beta\}\}$ (READ)
 $\{\{\exists v, \alpha, \beta. (\mathbf{x} \hookrightarrow_2 v, z * \text{list } \alpha z * \text{list } \beta \mathbf{y}) \wedge \alpha_0^\dagger = (v \cdot \alpha)^\dagger \cdot \beta\}\}$
[x + 4] := y;
 $\{\{\exists v, \alpha, \beta. (\mathbf{x} \hookrightarrow_2 v, \mathbf{y} * \text{list } \alpha z * \text{list } \beta \mathbf{y}) \wedge \alpha_0^\dagger = (v \cdot \alpha)^\dagger \cdot \beta\}\}$ (WRITE)
 $\{\{\exists v, \alpha, \beta. (\text{list } \alpha z * \text{list } (v \cdot \beta) \mathbf{x}) \wedge \alpha_0^\dagger = \alpha^\dagger \cdot v \cdot \beta\}\}$
 $\{\{\exists \alpha, \beta. (\text{list } \alpha z * \text{list } \beta \mathbf{x}) \wedge \alpha_0^\dagger = \alpha^\dagger \cdot \beta \wedge \underline{\text{defined}}(\mathbf{x})\}\}$
y := x;
 $\{\{\exists \alpha, \beta. (\text{list } \alpha z * \text{list } \beta \mathbf{y}) \wedge \alpha_0^\dagger = \alpha^\dagger \cdot \beta \wedge \underline{\text{defined}}(\mathbf{z})\}\}$ (ASSIGN)
x := z
 $\{\{\exists \alpha, \beta. (\text{list } \alpha \mathbf{x} * \text{list } \beta \mathbf{y}) \wedge \alpha_0^\dagger = \alpha^\dagger \cdot \beta \wedge \underline{\text{word}}(\mathbf{x} \neq 0)\}\}$ (ASSIGN)
od;
 $\{\{\exists \alpha, \beta. (\text{list } \alpha \mathbf{x} * \text{list } \beta \mathbf{y}) \wedge \alpha_0^\dagger = \alpha^\dagger \cdot \beta \wedge \mathbf{x} = 0\}\}$ (While)
 $\{\{\text{list } \alpha_0^\dagger \mathbf{y}\}\}$

4.7 Array Copy

$\{\{\mathbf{x} \hookrightarrow_n v_1, \dots, v_n\}\}$
y := ALLOC(ENC(n));
 $\{\{\mathbf{x} \hookrightarrow_n v_1, \dots, v_n * \mathbf{y} \hookrightarrow_n 0, \dots, 0\}\}$ (ALLOC)

$$\{(x \hookrightarrow_n v_1, \dots, v_n * y \hookrightarrow_n 0, \dots, 0) \wedge \underline{\text{safe}(x, y, t)}\} \quad (\text{Incl})$$

$$\{(x \hookrightarrow_n v_1, \dots, v_n * y \hookrightarrow_n 0, \dots, 0) \wedge x + 4n = x + 4n \wedge \underline{\text{safe}(x, y, t)} \wedge \underline{\text{defined}(x + 4n)}\}$$

$z := x + 4n;$

$$\{(x \hookrightarrow_n v_1, \dots, v_n * y \hookrightarrow_n 0, \dots, 0) \wedge z = x + 4n \wedge \underline{\text{safe}(x, y, t)}\} \quad (\text{Assign}')$$

$$\{\exists k. (x - 4k \hookrightarrow_n v_1, \dots, v_n * y - 4k \hookrightarrow_k v_1, \dots, v_k * y \hookrightarrow_{n-k} 0, \dots, 0) \wedge \\ 0 \leq k \leq n \wedge z = x + 4(n - k) \wedge \underline{\text{safe}(x - 4k, y - 4k, t)} \wedge \underline{\text{word}(x \neq z)}\}$$

while $x \neq z$ **do**

$$\{\exists k. (x - 4k \hookrightarrow_n v_1, \dots, v_n * y - 4k \hookrightarrow_k v_1, \dots, v_k * y \hookrightarrow_{n-k} 0, \dots, 0) \wedge \\ 0 \leq k \leq n \wedge z = x + 4(n - k) \wedge \underline{\text{safe}(x - 4k, y - 4k, t)} \wedge x \neq z\} \quad (\text{While})$$

$$\{\exists k. (x - 4k \hookrightarrow_n v_1, \dots, v_n * y - 4k \hookrightarrow_k v_1, \dots, v_k * y \hookrightarrow 0 * y + 4 \hookrightarrow_{n-(k+1)} 0, \dots, 0) \wedge \\ 0 \leq k < n \wedge z = x + 4(n - k) \wedge \underline{\text{safe}(x - 4k, y - 4k)}\}$$

$t := [x];$

$$\{\exists k. (x - 4k \hookrightarrow_n v_1, \dots, v_n * y - 4k \hookrightarrow_k v_1, \dots, v_k * y \hookrightarrow 0 * y + 4 \hookrightarrow_{n-(k+1)} 0, \dots, 0) \wedge \\ 0 \leq k < n \wedge z = x + 4(n - k) \wedge \underline{\text{safe}(x - 4k, y - 4k)} \wedge t = v_{k+1}\} \quad (\text{Read}')$$

$$\{\exists k. (x - 4k \hookrightarrow_n v_1, \dots, v_n * y - 4k \hookrightarrow_k v_1, \dots, v_k * y \hookrightarrow 0 * y + 4 \hookrightarrow_{n-(k+1)} 0, \dots, 0) \wedge \\ 0 \leq k < n \wedge z = x + 4(n - k) \wedge \underline{\text{safe}(x - 4k, y - 4k, t)} \wedge t = v_{k+1}\}$$

$[y] := t;$

$$\{\exists k. (x - 4k \hookrightarrow_n v_1, \dots, v_n * y - 4k \hookrightarrow_k v_1, \dots, v_k * y \hookrightarrow t * y + 4 \hookrightarrow_{n-(k+1)} 0, \dots, 0) \wedge \\ 0 \leq k < n \wedge z = x + 4(n - k) \wedge \underline{\text{safe}(x - 4k, y - 4k)} \wedge t = v_{k+1}\} \quad (\text{Write})$$

$$\{\exists k. (x - 4k \hookrightarrow_n v_1, \dots, v_n * y - 4k \hookrightarrow_{k+1} v_1, \dots, v_k, v_{k+1} * y + 4 \hookrightarrow_{n-(k+1)} 0, \dots, 0) \wedge \\ 0 \leq k < n \wedge z = x + 4(n - k) \wedge \underline{\text{safe}(x - 4k, y - 4k, t)} \wedge \underline{\text{defined}(x + 4)}\}$$

$x := x + 4;$

$$\{\exists k. (x - 4(k + 1) \hookrightarrow_n v_1, \dots, v_n * y - 4k \hookrightarrow_{k+1} v_1, \dots, v_{k+1} * y + 4 \hookrightarrow_{n-(k+1)} 0, \dots, 0) \wedge \\ 0 \leq k < n \wedge z = x + 4(n - (k + 1)) \wedge \underline{\text{safe}(x - 4(k + 1), y - 4k, t)} \wedge \underline{\text{defined}(y + 4)}\} \quad (\text{Assign}')$$

$y := y + 4;$

$$\{\exists k. (x - 4(k + 1) \hookrightarrow_n v_1, \dots, v_n * y - 4(k + 1) \hookrightarrow_{k+1} v_1, \dots, v_{k+1} * y \hookrightarrow_{n-(k+1)} 0, \dots, 0) \wedge \\ 0 \leq k < n \wedge z = x + 4(n - (k + 1)) \wedge \underline{\text{safe}(x - 4(k + 1), y - 4(k + 1), t)}\} \quad (\text{Assign}')$$

$$\{\exists k. (x - 4k \hookrightarrow_n v_1, \dots, v_n * y - 4k \hookrightarrow_k v_1, \dots, v_k * y \hookrightarrow_{n-k} 0, \dots, 0) \wedge \\ 0 \leq k \leq n \wedge z = x + 4(n - k) \wedge \underline{\text{safe}(x - 4k, y - 4k, t)} \wedge \underline{\text{word}(x \neq z)}\}$$

od;

$$\{\exists k. (x - 4k \hookrightarrow_n v_1, \dots, v_n * y - 4k \hookrightarrow_k v_1, \dots, v_k * y \hookrightarrow_{n-k} 0, \dots, 0) \wedge \\ 0 \leq k \leq n \wedge z = x + 4(n - k) \wedge \underline{\text{safe}(x - 4k, y - 4k, t)} \wedge x = z\} \quad (\text{While})$$

$$\{(x - 4n \hookrightarrow_n v_1, \dots, v_n * y - 4n \hookrightarrow_n v_1, \dots, v_n) \wedge \underline{\text{safe}(x - 4n, y - 4n, t)}\}$$

$x := x - 4n;$

$\{(x \hookrightarrow_n v_1, \dots, v_n * y - 4n \hookrightarrow_n v_1, \dots, v_n) \wedge \underline{\text{safe}(x, y - 4n, t)}\}$ (Assign')

$y := y - 4n;$

$\{(x \hookrightarrow_n v_1, \dots, v_n * y \hookrightarrow_n v_1, \dots, v_n) \wedge \underline{\text{safe}(x, y, t, 0)}\}$ (Assign')

$z := 0$

$\{(x \hookrightarrow_n v_1, \dots, v_n * y \hookrightarrow_n v_1, \dots, v_n) \wedge \underline{\text{safe}(x, y, t, z)}\}$ (Assign')

$\{(x \hookrightarrow_n v_1, \dots, v_n * y \hookrightarrow_n v_1, \dots, v_n)\}$ (Incl)

5 Soundness of Program Logic

5.1 Basic Lemmas

Lemma 1.

$$\llbracket \text{defined}(\mathbf{E}) \rrbracket_s \in \text{Words} \setminus \{0\} \quad \text{iff} \quad \llbracket \mathbf{E} \rrbracket_s \neq \text{undef}$$

Proof. By a case analysis on $\llbracket \mathbf{E} \rrbracket_s$: when $\llbracket \mathbf{E} \rrbracket_s \in \text{LogVals}$, we have $\llbracket \mathbf{E} = \mathbf{E} \rrbracket_s = 1 \in \text{Words} \setminus \{0\}$; when $\llbracket \mathbf{E} \rrbracket_s = \text{undef}$, we have $\llbracket \mathbf{E} = \mathbf{E} \rrbracket_s = \text{undef} \notin \text{Words} \setminus \{0\}$. \square

Lemma 2.

$$s \sim_{\mathbf{T}} s \wedge \llbracket E \rrbracket_s \neq \text{undef} \implies \llbracket E \rrbracket_s = \text{phyv}_{\mathbf{T}}(\llbracket E \rrbracket_s) \neq \text{undef}$$

Proof. It can be shown by induction over E .

- When $E = x$:
From $s \sim_{\mathbf{T}} s$, we have $\text{phyv}_{\mathbf{T}}(s(x)) = s(x) \neq \text{undef}$.
- When $E = w$:
 $\text{phyv}_{\mathbf{T}}(\llbracket w \rrbracket_s) = \text{phyv}_{\mathbf{T}}(w) = w = \llbracket w \rrbracket_s \neq \text{undef}$.
- When $E = (E_1 \star E_2)$:
From $\llbracket E \rrbracket_s \neq \text{undef}$, we have the following cases:
 - When $\llbracket E_1 \rrbracket_s = w_1 \in \text{Words} \wedge \llbracket E_2 \rrbracket_s = w_2 \in \text{Words} \wedge \llbracket E \rrbracket_s = w_1 \star w_2 \neq \text{undef}$:
By induction hypothesis, we have $\llbracket E_k \rrbracket_s = \text{phyv}_{\mathbf{T}}(w_k) = w_k$ for $k = 1, 2$.
Thus, we have $\text{phyv}_{\mathbf{T}}(\llbracket E \rrbracket_s) = \text{phyv}_{\mathbf{T}}(w_1 \star w_2) = w_1 \star w_2 = \llbracket E \rrbracket_s \neq \text{undef}$.
 - When $\star = + \wedge \llbracket E_k \rrbracket_s = \ell \hat{+} i \wedge \llbracket E_{3-k} \rrbracket_s = w \wedge \llbracket E \rrbracket_s = \ell \hat{+} (i + w)$:
By induction hypothesis, we have $\llbracket E_k \rrbracket_s = \text{phyv}_{\mathbf{T}}(\ell \hat{+} i) = p + i$ for $(p, n) = \mathbf{T}(\ell)$; and $\llbracket E_{3-k} \rrbracket_s = \text{phyv}_{\mathbf{T}}(w) = w$.
So we have $\text{phyv}_{\mathbf{T}}(\llbracket E \rrbracket_s) = \text{phyv}_{\mathbf{T}}(\ell \hat{+} (i + w)) = p + (i + w) = (p + i) + w = \llbracket E \rrbracket_s \neq \text{undef}$.
 - When $\star = - \wedge \llbracket E_1 \rrbracket_s = \ell \hat{+} i \wedge \llbracket E_2 \rrbracket_s = w \wedge \llbracket E \rrbracket_s = \ell \hat{+} (i - w)$:
By induction hypothesis, we have $\llbracket E_1 \rrbracket_s = \text{phyv}_{\mathbf{T}}(\ell \hat{+} i) = p + i$ for $(p, n) = \mathbf{T}(\ell)$; and $\llbracket E_2 \rrbracket_s = \text{phyv}_{\mathbf{T}}(w) = w$.
So we have $\text{phyv}_{\mathbf{T}}(\llbracket E \rrbracket_s) = \text{phyv}_{\mathbf{T}}(\ell \hat{+} (i - w)) = p + (i - w) = (p + i) - w = \llbracket E \rrbracket_s \neq \text{undef}$.
 - When $\star = - \wedge \llbracket E_1 \rrbracket_s = \ell \hat{+} i \wedge \llbracket E_2 \rrbracket_s = \ell \hat{+} j \wedge \llbracket E \rrbracket_s = i - j$:
By induction hypothesis, we have $\llbracket E_1 \rrbracket_s = \text{phyv}_{\mathbf{T}}(\ell \hat{+} i) = p + i$ and $\llbracket E_2 \rrbracket_s = \text{phyv}_{\mathbf{T}}(\ell \hat{+} j) = p + j$ for $(p, n) = \mathbf{T}(\ell)$.
So we have $\text{phyv}_{\mathbf{T}}(\llbracket E \rrbracket_s) = \text{phyv}_{\mathbf{T}}(i - j) = i - j = (p + i) - (p + j) = \llbracket E \rrbracket_s \neq \text{undef}$.
 - When $\star = < \wedge \llbracket E_1 \rrbracket_s = \ell \hat{+} i \wedge \llbracket E_2 \rrbracket_s = \ell \hat{+} j \wedge \llbracket E \rrbracket_s = i < j$:
By induction hypothesis, we have $\llbracket E_1 \rrbracket_s = \text{phyv}_{\mathbf{T}}(\ell \hat{+} i) = p + i$ and $\llbracket E_2 \rrbracket_s = \text{phyv}_{\mathbf{T}}(\ell \hat{+} j) = p + j$ for $(p, n) = \mathbf{T}(\ell)$.
So we have $\text{phyv}_{\mathbf{T}}(\llbracket E \rrbracket_s) = \text{phyv}_{\mathbf{T}}(i < j) = i < j = (p + i) < (p + j) = \llbracket E \rrbracket_s \neq \text{undef}$.
 - When $\star = = \wedge \llbracket E_1 \rrbracket_s = \ell \hat{+} i \wedge \llbracket E_2 \rrbracket_s = \ell \hat{+} j \wedge \llbracket E \rrbracket_s = (i = j)$:
By induction hypothesis, we have $\llbracket E_1 \rrbracket_s = \text{phyv}_{\mathbf{T}}(\ell \hat{+} i) = p + i$ and $\llbracket E_2 \rrbracket_s = \text{phyv}_{\mathbf{T}}(\ell \hat{+} j) = p + j$ for $(p, n) = \mathbf{T}(\ell)$.
So we have $\text{phyv}_{\mathbf{T}}(\llbracket E \rrbracket_s) = \text{phyv}_{\mathbf{T}}(i = j) = (i = j) = ((p + i) = (p + j)) = \llbracket E \rrbracket_s \neq \text{undef}$.

- When $\star = = \wedge \llbracket E_1 \rrbracket_s = \ell \hat{+} 0 \wedge \llbracket E_2 \rrbracket_s = \ell' \hat{+} 0 \wedge \llbracket E \rrbracket_s = (\ell = \ell')$:
 By induction hypothesis, we have $\llbracket E_1 \rrbracket_s = \text{phyv}_{\mathbf{T}}(\ell \hat{+} 0) = p$ and $\llbracket E_2 \rrbracket_s = \text{phyv}_{\mathbf{T}}(\ell' \hat{+} 0) = p'$ for $(p, n) = \mathbf{T}(\ell)$ and $(p', n') = \mathbf{T}(\ell')$.
 So we have $\text{phyv}_{\mathbf{T}}(\llbracket E \rrbracket_s) = \text{phyv}_{\mathbf{T}}(\ell = \ell') = (\ell = \ell') = (p = p') = \llbracket E \rrbracket_s \neq \text{undef}$.
- When $\star = = \wedge \llbracket E_k \rrbracket_s = \ell \hat{+} 4i \wedge i \geq 0 \wedge \llbracket E_{3-k} \rrbracket_s \in \text{NonPtrs} \wedge \llbracket E \rrbracket_s = 0$
 By induction hypothesis, we have $\llbracket E_k \rrbracket_s = \text{phyv}_{\mathbf{T}}(\ell \hat{+} 4i) = p + 4i \in \text{Ptrs}$ for $(p, n) = \mathbf{T}(\ell)$, and $\llbracket E_{3-k} \rrbracket_s = \text{phyv}_{\mathbf{T}}(\llbracket E_k \rrbracket_s) = \llbracket E_k \rrbracket_s \in \text{NonPtrs}$.
 So we have $\text{phyv}_{\mathbf{T}}(\llbracket E \rrbracket_s) = \text{phyv}_{\mathbf{T}}(0) = 0 = (\llbracket E_k \rrbracket_s = \llbracket E_{3-k} \rrbracket_s) = \llbracket E \rrbracket_s \neq \text{undef}$.
- When $E = \text{not } E'$:
 From $\llbracket E \rrbracket_s \neq \text{undef}$, we have $\llbracket E' \rrbracket_s = w \in \text{Words} \wedge \llbracket E \rrbracket_s = \text{not } w$.
 By induction hypothesis, we have $\llbracket E' \rrbracket_s = \text{phyv}_{\mathbf{T}}(w) = w$.
 Thus, we have $\text{phyv}_{\mathbf{T}}(\llbracket E \rrbracket_s) = \text{phyv}_{\mathbf{T}}(\text{not } w) = \text{not } w = \llbracket E \rrbracket_s \neq \text{undef}$.

□

Corollary 3.

$$\mathbf{s} \sim_{\mathbf{T}} s \wedge \llbracket E \rrbracket_s = \ell \hat{+} i \implies \ell \in \text{dom}(\mathbf{T})$$

Proof. By Lemma 2, we have $\text{phyv}_{\mathbf{T}}(\ell \hat{+} i) \neq \text{undef}$, from which it follows that $\ell \in \text{dom}(\mathbf{T})$. □

Lemma 4. When $\llbracket E' \rrbracket_s \neq \text{undef}$,

- (1) $\llbracket \mathbf{E}[E'/x] \rrbracket_s = \llbracket \mathbf{E} \rrbracket_{(\mathbf{s} \mid x \mapsto \llbracket E' \rrbracket_s)}$
- (2) $\mathbf{s}, \mathbf{h} \models_{\mathbf{L}} \mathbf{P}[E'/x] \quad \text{iff} \quad (\mathbf{s} \mid x \mapsto \llbracket E' \rrbracket_s), \mathbf{h} \models_{\mathbf{L}} \mathbf{P}$

Proof. (1) can be shown by an induction on \mathbf{E} . When $\mathbf{E} = y$: if $y = x$, then both LHS and RHS are equal to $\llbracket E' \rrbracket_s$; otherwise, both are equal to $\mathbf{s}(y)$. The other cases are straightforward.

(2) follows from (1) by a simple induction on \mathbf{P} . □

Lemma 5.

- (1) $(\forall x \in \text{FPV}(\mathbf{E}). \mathbf{s}(x) = \mathbf{s}'(x)) \implies \llbracket \mathbf{E} \rrbracket_s = \llbracket \mathbf{E} \rrbracket_{s'}$
- (2) $(\forall x \in \text{FPV}(\mathbf{P}). \mathbf{s}(x) = \mathbf{s}'(x)) \implies (\mathbf{s}, \mathbf{h} \models_{\mathbf{L}} \mathbf{P} \iff \mathbf{s}', \mathbf{h} \models_{\mathbf{L}} \mathbf{P})$

Proof.

(1) : By a simple induction on \mathbf{E} .

(2) : By a simple induction on \mathbf{P} using (1).

□

Lemma 6. $\mathbf{s}, \mathbf{h} \models_{\mathbf{L}} P \iff \mathbf{s}, \mathbf{h} \models P$

Proof. By a simple induction on P . □

Lemma 7.

$$\mathbf{s}, \mathbf{h} \models_{\mathbf{L}} (\exists v. \mathbf{P})[\rho] \iff \exists v \in \text{LogVals}. \mathbf{s}, \mathbf{h} \models_{\mathbf{L}} \mathbf{P}[(\rho \mid v \mapsto \mathbf{v})]$$

Proof. Choose a fresh $u \notin \text{dom}(\rho)$. Then the goal follows from

- $(\exists v. \mathbf{P})[\rho] \approx_\alpha (\exists u. \mathbf{P}[u/v])[\rho] = \exists u. \mathbf{P}[u/v][\rho]$,
- $\mathbf{s}, \mathbf{h} \models_{\mathbf{L}} \exists u. \mathbf{P}[u/v][\rho] \iff \exists \mathbf{v} \in \text{LogVals}. \mathbf{s}, \mathbf{h} \models_{\mathbf{L}} \mathbf{P}[u/v][\rho][\mathbf{v}/u]$,
- $\mathbf{P}[u/v][\rho][\mathbf{v}/u] = \mathbf{P}[u/v][\mathbf{v}/u][\rho] = \mathbf{P}[\mathbf{v}/v][\rho] = \mathbf{P}[(\rho \mid v \mapsto \mathbf{v})]$.

□

Lemma 8.

$$R \subseteq R' \wedge h \subseteq h' \wedge \sigma \subseteq \sigma' \implies \text{reach}(R, h, \sigma) \subseteq \text{reach}(R', h', \sigma')$$

Proof. One can easily show that $\text{reach}_n(R, h, \sigma) \subseteq \text{reach}_n(R', h', \sigma')$ by induction on n . □

Lemma 9. When $\overline{\text{dom}}(\sigma) \subseteq \text{dom}(h)$ and $\sigma \subseteq \sigma'$,

$$\text{reach}(\text{dom}(\sigma), h, \sigma') \subseteq \text{dom}(\sigma) \iff \forall p \in \overline{\text{dom}}(\sigma). h(p) \in \text{dom}(\sigma) \cup \text{NonPtrs}$$

Proof.

- \implies part:
Let $p \in \overline{\text{dom}}(\sigma)$. As $\overline{\text{dom}}(\sigma) \subseteq \text{dom}(h)$, we have $h(p) \in \text{Words}$.
If $h(p) \in \text{NonPtrs}$, then trivially $h(p) \in \text{dom}(\sigma) \cup \text{NonPtrs}$.
If $h(p) \in \text{Ptrs}$, then $h(p) \in \text{reach}_1(\text{dom}(\sigma), h, \sigma') \subseteq \text{dom}(\sigma) \subseteq \text{dom}(\sigma) \cup \text{NonPtrs}$.
- \Leftarrow part:
We prove $\text{reach}_n(\text{dom}(\sigma), h, \sigma') \subseteq \text{dom}(\sigma)$ by induction on n .
Base case: $\text{reach}_0(\text{dom}(\sigma), h, \sigma') = \text{dom}(\sigma) \subseteq \text{dom}(\sigma)$.
Inductive step: $\text{reach}_{n+1}(\text{dom}(\sigma), h, \sigma') \subseteq \text{dom}(\sigma)$ directly follows from
(1) the induction hypothesis: $\text{reach}_n(\text{dom}(\sigma), h, \sigma') \subseteq \text{dom}(\sigma)$; and
(2) the fact that $\forall p \in \overline{\text{dom}}(\sigma). h(p) \in \text{Ptrs} \implies h(p) \in \text{dom}(\sigma)$.

□

Lemma 10.

$$\mathbf{h} \sim_{\mathbf{T}} h \wedge \mathbf{h} :: \mathbf{T} \wedge \sigma \supseteq \text{shape}(\mathbf{T}) \implies \text{reach}(\text{dom}(\text{shape}(\mathbf{T})), h, \sigma) \subseteq \text{dom}(\text{shape}(\mathbf{T}))$$

Proof.

- Assume: $\mathbf{h} \approx_{\mathbf{T}} h$ and let $\sigma \supseteq \text{shape}(\mathbf{T})$.
- As $\text{phyh}_{\mathbf{T}}(\mathbf{h}) \subseteq h$, we have $\overline{\text{dom}}(\text{shape}(\mathbf{T})) = \text{dom}(\text{phyh}_{\mathbf{T}}(\mathbf{h})) \subseteq \text{dom}(h)$.
- To show: $\text{reach}(\text{dom}(\text{shape}(\mathbf{T})), h, \sigma) \subseteq \text{dom}(\text{shape}(\mathbf{T}))$.
- By Lemma 9, it suffices to show that $\forall p \in \overline{\text{dom}}(\text{shape}(\mathbf{T})). h(p) \in \text{dom}(\text{shape}(\mathbf{T})) \cup \text{NonPtrs}$.
- Let $p \in \overline{\text{dom}}(\text{shape}(\mathbf{T}))$.
Since $\text{phyh}_{\mathbf{T}}(\mathbf{h}) \subseteq h$, there exists ℓ', p', n', i such that $(p', n') = \mathbf{T}(\ell') \wedge i < n' \wedge p = p' + 4i \wedge h(p) = \text{phyv}_{\mathbf{T}}(\mathbf{h}(\ell')(i))$.
- From $\mathbf{h} :: \mathbf{T}$, it follows that $\mathbf{h}(\ell')(i) \in \text{Safe}(\text{dom}(\mathbf{T}))$. Thus, we have two cases.

- When $\mathbf{h}(\ell')(i) \in \text{NonPtrs}$:
 $h(p) = \text{phyv}_{\mathbf{T}}(\mathbf{h}(\ell')(i)) = \mathbf{h}(\ell')(i) \in \text{NonPtrs} \subseteq \text{dom}(\text{shape}(\mathbf{T})) \cup \text{NonPtrs}$.
- When $\mathbf{h}(\ell')(i) = \ell'' \hat{+} 0$ for $\ell'' \in \text{dom}(\mathbf{T})$:
 $h(p) = \text{phyv}_{\mathbf{T}}(\ell'' \hat{+} 0) = p''$ for $(p'', n'') = \mathbf{T}(\ell'')$.
Thus, $h(p) \in \text{dom}(\text{shape}(\mathbf{T})) \subseteq \text{dom}(\text{shape}(\mathbf{T})) \cup \text{NonPtrs}$.

□

5.2 Soundness of Inner-level Rules

Definition 1 (Generalized triple).

$$\begin{aligned} \{\mathbf{P}\} C \{\mathbf{Q}\} : k \text{ iff } & \forall j \leq k. \forall \rho \in \text{Env}(\text{FLV}(\mathbf{P}, \mathbf{Q})), \mathbf{s}, \mathbf{h}, \mathbf{h}_{\mathbf{F}}, \mathbf{T}, s, h, C', s', h'. \\ & \mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} \mathbf{P}[\rho] \wedge \mathbf{s} \sim_{\mathbf{T}} s \wedge \mathbf{h} \uplus \mathbf{h}_{\mathbf{F}} \approx_{\mathbf{T}} h \wedge C, s, h \rightsquigarrow^j C', s', h' \implies \\ & (C', s', h' \rightsquigarrow -) \vee \\ & (\exists \mathbf{s}', \mathbf{h}'. C' = \text{skip} \wedge \mathbf{s}', \mathbf{h}' \models_{\text{dom}(\mathbf{T})} \mathbf{Q}[\rho] \wedge \\ & (\forall \mathbf{x} \notin \text{Mod}(C). \mathbf{s}'(\mathbf{x}) = \mathbf{s}(\mathbf{x})) \wedge \mathbf{s}' \sim_{\mathbf{T}} s' \wedge \mathbf{h}' \uplus \mathbf{h}_{\mathbf{F}} \approx_{\mathbf{T}} h') \end{aligned}$$

5.2.1 Skip

Theorem 1 (Soundness: Skip).

$$\frac{}{[\text{true}] \text{ skip } [\text{true}]}$$

Proof.

- Assume: $\mathbf{s}, \mathbf{h}, \mathbf{h}_{\mathbf{F}}, \mathbf{T}, s, h, C', s', h'$ such that
 $\checkmark \mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} \text{true} \wedge \mathbf{s} \sim_{\mathbf{T}} s \wedge \mathbf{h} \uplus \mathbf{h}_{\mathbf{F}} \approx_{\mathbf{T}} h \wedge \text{skip}, s, h \rightsquigarrow^* C', s', h'$
- skip, s, h does not diverge as it takes no step.
- To show:
(*) $C', s', h' \rightsquigarrow -$; or
(**) $\exists \mathbf{s}', \mathbf{h}'. C' = \text{skip} \wedge \mathbf{s}', \mathbf{h}' \models_{\text{dom}(\mathbf{T})} \text{true} \wedge$
 $(\forall \mathbf{y} \notin \text{Mod}(C). \mathbf{s}'(\mathbf{y}) = \mathbf{s}(\mathbf{y})) \wedge \mathbf{s}' \sim_{\mathbf{T}} s' \wedge \mathbf{h}' \uplus \mathbf{h}_{\mathbf{F}} \approx_{\mathbf{T}} h'$
- From $\text{skip}, s, h \rightsquigarrow^* C', s', h'$, we have $C' = \text{skip}$, $s' = s$ and $h' = h$.
- (**) holds by letting $\mathbf{s}' = \mathbf{s}$ and $\mathbf{h}' = \mathbf{h}$.

□

5.2.2 Assign

Theorem 2 (Soundness: Assign).

$$\frac{}{[\mathbf{x} = v \wedge \text{defined}(E)] \mathbf{x} := E \ [\mathbf{x} = E[v/\mathbf{x}]}$$

Proof.

- Substitute the logical variables v with an arbitrary logical words \mathbf{v} .

- Assume: $\mathbf{s}, \mathbf{h}, \mathbf{h}_F, \mathbf{T}, s, h, C', s', h'$ such that
 - ✓ $\mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} (\mathbf{x} = \mathbf{v} \wedge \text{defined}(E)) \wedge \mathbf{s} \sim_{\mathbf{T}} s \wedge \mathbf{h} \uplus \mathbf{h}_F \approx_{\mathbf{T}} h \wedge \mathbf{x} := E, s, h \rightsquigarrow^* C', s', h'$
- $\mathbf{x} := E, s, h$ does not diverge as it takes at most one step.
- To show:
 - (*) $C', s', h' \rightsquigarrow -$; or
 - (**) $\exists \mathbf{s}', \mathbf{h}'. C' = \text{skip} \wedge \mathbf{s}', \mathbf{h}' \models_{\text{dom}(\mathbf{T})} (\mathbf{x} = E[\mathbf{v}/\mathbf{x}]) \wedge$
 $(\forall \mathbf{y} \notin \text{Mod}(C). \mathbf{s}'(\mathbf{y}) = \mathbf{s}(\mathbf{y})) \wedge \mathbf{s}' \sim_{\mathbf{T}} s' \wedge \mathbf{h}' \uplus \mathbf{h}_F \approx_{\mathbf{T}} h'$
- As $\mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} (\mathbf{x} = \mathbf{v} \wedge \text{defined}(E))$ and $\mathbf{s} \sim_{\mathbf{T}} s$, by Lemmas 1 and 2 we have
 - ✓ $\mathbf{s}(\mathbf{x}) = \mathbf{v}$
 - ✓ $\llbracket E \rrbracket_{\mathbf{s}} = \text{phyv}_{\mathbf{T}}(\llbracket E \rrbracket_{\mathbf{s}}) \neq \text{undef}$
- From $\mathbf{x} := E, s, h \rightsquigarrow^* C', s', h'$, we have the following two cases:
 - When $C' = (\mathbf{x} := E) \wedge s' = s \wedge h' = h$:
 As $\llbracket E \rrbracket_{\mathbf{s}} \neq \text{undef}$, it follows that $\mathbf{x} := E, s, h \rightsquigarrow \text{skip}, (s \mid \mathbf{x} \mapsto \llbracket E \rrbracket_{\mathbf{s}}), h$. Thus, (*) holds.
 - When $C' = \text{skip} \wedge s' = (s \mid \mathbf{x} \mapsto \llbracket E \rrbracket_{\mathbf{s}}) \wedge h' = h$:
 (**) holds by letting $\mathbf{s}' = (s \mid \mathbf{x} \mapsto \llbracket E \rrbracket_{\mathbf{s}})$ and $\mathbf{h}' = \mathbf{h}$ because
 - $\mathbf{s}', \mathbf{h}' \models_{\text{dom}(\mathbf{T})} (\mathbf{x} = E[\mathbf{v}/\mathbf{x}])$ follows from

$$\mathbf{s}'(\mathbf{x}) = \llbracket E \rrbracket_{\mathbf{s}} = \llbracket E \rrbracket_{(s' \mid \mathbf{x} \mapsto \mathbf{v})} = \llbracket E[\mathbf{v}/\mathbf{x}] \rrbracket_{\mathbf{s}'} \neq \text{undef},$$
 which holds by Lemmas 5 and 4 as $\mathbf{s}(\mathbf{x}) = \mathbf{v}$; and
 - $\mathbf{s}' \sim_{\mathbf{T}} s'$ holds since $\mathbf{s} \sim_{\mathbf{T}} s$ and $\llbracket E \rrbracket_{\mathbf{s}} = \text{phyv}_{\mathbf{T}}(\llbracket E \rrbracket_{\mathbf{s}})$.

□

5.2.3 Read

Theorem 3 (Soundness: Read).

$$\overline{[\mathbf{x} = u \wedge E \hookrightarrow v] \ \mathbf{x} := [E] \ [\mathbf{x} = v \wedge E[u/\mathbf{x}] \hookrightarrow v]}$$

Proof.

- Substitute the logical variables u, v with two arbitrary logical words $\mathbf{v}_1, \mathbf{v}_2$.
- Assume: $\mathbf{s}, \mathbf{h}, \mathbf{h}_F, \mathbf{T}, s, h, C', s', h'$ such that
 - ✓ $\mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} (\mathbf{x} = \mathbf{v}_1 \wedge E \hookrightarrow \mathbf{v}_2) \wedge \mathbf{s} \sim_{\mathbf{T}} s \wedge \mathbf{h} \uplus \mathbf{h}_F \approx_{\mathbf{T}} h \wedge \mathbf{x} := [E], s, h \rightsquigarrow^* C', s', h'$
- $\mathbf{x} := [E], s, h$ does not diverge as it takes at most one step.
- To show:
 - (*) $C', s', h' \rightsquigarrow -$; or
 - (**) $\exists \mathbf{s}', \mathbf{h}'. C' = \text{skip} \wedge \mathbf{s}', \mathbf{h}' \models_{\text{dom}(\mathbf{T})} (\mathbf{x} = \mathbf{v}_2 \wedge E[\mathbf{v}_1/\mathbf{x}] \hookrightarrow \mathbf{v}_2) \wedge$
 $(\forall \mathbf{y} \notin \text{Mod}(C). \mathbf{s}'(\mathbf{y}) = \mathbf{s}(\mathbf{y})) \wedge \mathbf{s}' \sim_{\mathbf{T}} s' \wedge \mathbf{h}' \uplus \mathbf{h}_F \approx_{\mathbf{T}} h'$

- From $\mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} (\mathbf{x} = \mathbf{v}_1 \wedge E \hookrightarrow \mathbf{v}_2)$, by Corollary 3 we have
 $\checkmark \mathbf{s}(\mathbf{x}) = \mathbf{v}_1 \wedge \llbracket E \rrbracket_{\mathbf{s}} = \widehat{\ell+4i} \wedge \mathbf{h}(\ell)(i) = \mathbf{v}_2 \wedge \ell \in \text{dom}(\mathbf{T})$.
- As $\mathbf{s} \sim_{\mathbf{T}} s$ and $\llbracket E \rrbracket_{\mathbf{s}} = \widehat{\ell+4i}$, by Lemma 2 we have
 $\checkmark \llbracket E \rrbracket_s = \text{phyv}_{\mathbf{T}}(\widehat{\ell+4i}) = p + 4i$ for $(p, n) = \mathbf{T}(\ell)$.
- From $\mathbf{h} \uplus \mathbf{h}_{\mathbf{F}} : \mathbf{T} \wedge (p, n) = \mathbf{T}(\ell) \wedge \mathbf{h}(\ell)(i) \neq \text{undef}$, we have
 $\checkmark i < n$.
- From $\text{phyh}_{\mathbf{T}}(\mathbf{h} \uplus \mathbf{h}_{\mathbf{F}}) \subseteq h \wedge (p, n) = \mathbf{T}(\ell) \wedge i < n$, we have
 $\checkmark h(p + 4i) = \text{phyv}_{\mathbf{T}}(\mathbf{h}(\ell)(i)) \neq \text{undef}$.
- From $\mathbf{x} := [E], s, h \rightsquigarrow^* C', s', h'$, we have the following two cases:
- When $C' = (\mathbf{x} := [E]) \wedge s' = s \wedge h' = h$:
As $\llbracket E \rrbracket_s = p + 4i \wedge h(p + 4i) \neq \text{undef}$, we have $\mathbf{x} := [E], s, h \rightsquigarrow \text{skip}, (s \mid \mathbf{x} \mapsto h(p + 4i)), h$.
Thus, (*) holds.
- When $C' = \text{skip} \wedge s' = (s \mid \mathbf{x} \mapsto h(p + 4i)) \wedge h' = h$:
(**) holds by letting $\mathbf{s}' = (\mathbf{s} \mid \mathbf{x} \mapsto \mathbf{h}(\ell)(i))$ and $\mathbf{h}' = \mathbf{h}$ because
 - $\mathbf{s}' \sim_{\mathbf{T}} s'$ holds since $\mathbf{s} \sim_{\mathbf{T}} s$ and $h(p + 4i) = \text{phyv}_{\mathbf{T}}(\mathbf{h}(\ell)(i))$;
 - $\mathbf{s}', \mathbf{h}' \models_{\text{dom}(\mathbf{T})} \mathbf{x} = \mathbf{v}_2$ holds since $\mathbf{s}'(\mathbf{x}) = \mathbf{h}(\ell)(i) = \mathbf{v}_2$; and
 - $\mathbf{s}', \mathbf{h}' \models_{\text{dom}(\mathbf{T})} E[\mathbf{v}_1/\mathbf{x}] \hookrightarrow \mathbf{v}_2$ follows from
 - (1) $\llbracket E[\mathbf{v}_1/\mathbf{x}] \rrbracket_{\mathbf{s}'}$ = $\llbracket E \rrbracket_{(\mathbf{s}' \mid \mathbf{x} \mapsto \mathbf{v}_1)}$ (by Lemma 4)
= $\llbracket E \rrbracket_{(\mathbf{s} \mid \mathbf{x} \mapsto \mathbf{v}_1)}$ (as $(\mathbf{s}' \mid \mathbf{x} \mapsto \mathbf{v}_1) = (\mathbf{s} \mid \mathbf{x} \mapsto \mathbf{v}_1)$)
= $\llbracket E \rrbracket_{\mathbf{s}}$ (by Lemma 5, as $\mathbf{s}(\mathbf{x}) = \mathbf{v}_1$)
= $\widehat{\ell+4i}$,
 - (2) $\mathbf{h}'(\ell)(i) = \mathbf{h}(\ell)(i) = \mathbf{v}_2 \neq \text{undef}$.

□

5.2.4 Write

Theorem 4 (Soundness: Write).

$$\overline{[E \hookrightarrow - \wedge \text{safe}(E')] [E] := E' [E \hookrightarrow E']}$$

Proof.

- Assume: $\mathbf{s}, \mathbf{h}, \mathbf{h}_{\mathbf{F}}, \mathbf{T}, s, h, C', s', h'$ such that
 $\checkmark \mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} (E \hookrightarrow - \wedge \text{safe}(E')) \wedge \mathbf{s} \sim_{\mathbf{T}} s \wedge \mathbf{h} \uplus \mathbf{h}_{\mathbf{F}} \approx_{\mathbf{T}} h \wedge [E] := E', s, h \rightsquigarrow^* C', s', h'$
- $[E] := E', s, h$ does not diverge as it takes at most one step.
- To show:
 - (*) $C', s', h' \rightsquigarrow -$; or
 - (**) $\exists \mathbf{s}', \mathbf{h}'$. $C' = \text{skip} \wedge \mathbf{s}', \mathbf{h}' \models_{\text{dom}(\mathbf{T})} E \hookrightarrow E' \wedge$
 $(\forall \mathbf{y} \notin \text{Mod}(C). \mathbf{s}'(\mathbf{y}) = \mathbf{s}(\mathbf{y})) \wedge \mathbf{s}' \sim_{\mathbf{T}} s' \wedge \mathbf{h}' \uplus \mathbf{h}_{\mathbf{F}} \approx_{\mathbf{T}} h'$

- From $\mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} (E \hookrightarrow - \wedge \text{safe}(E'))$, by Corollary 3 we have
 $\checkmark \llbracket E \rrbracket_{\mathbf{s}} = \ell \hat{+} 4i \wedge \mathbf{h}(\ell)(i) \neq \text{undef} \wedge \ell \in \text{dom}(\mathbf{T}) \wedge \llbracket E' \rrbracket_{\mathbf{s}} \in \text{Safe}(\text{dom}(\mathbf{T}))$.
- As $\mathbf{s} \sim_{\mathbf{T}} s$ and $\llbracket E \rrbracket_{\mathbf{s}} = \ell \hat{+} 4i$, by Lemma 2 we have
 $\checkmark \llbracket E \rrbracket_s = \text{phyv}_{\mathbf{T}}(\ell \hat{+} 4i) = p + 4i$ for $(p, n) = \mathbf{T}(\ell)$.
- As $\mathbf{s} \sim_{\mathbf{T}} s$ and $\llbracket E' \rrbracket_{\mathbf{s}} \neq \text{undef}$, by Lemma 2 we have
 $\checkmark \llbracket E' \rrbracket_s = \text{phyv}_{\mathbf{T}}(\llbracket E' \rrbracket_{\mathbf{s}}) \neq \text{undef}$.
- From $\mathbf{h} \uplus \mathbf{h}_{\mathbf{F}} : \mathbf{T} \wedge (p, n) = \mathbf{T}(\ell) \wedge \mathbf{h}(\ell)(i) \neq \text{undef}$, we have
 $\checkmark i < n$.
- From $\text{phyh}_{\mathbf{T}}(\mathbf{h} \uplus \mathbf{h}_{\mathbf{F}}) \subseteq h \wedge (p, n) = \mathbf{T}(\ell) \wedge i < n \wedge \mathbf{h}(\ell)(i) \neq \text{undef}$, we have
 $\checkmark \text{phyv}_{\mathbf{T}}(\mathbf{h}(\ell)(i)) = h(p + 4i) \neq \text{undef}$.
- From $[E] := E', s, h \rightsquigarrow^* C', s', h'$, we have the following two cases:
- When $C' = ([E] := E') \wedge s' = s \wedge h' = h$:
As $\llbracket E \rrbracket_s = p + 4i \wedge h(p + 4i) \neq \text{undef} \wedge \llbracket E' \rrbracket_s \neq \text{undef}$, we have

$$[E] := E', s, h \rightsquigarrow \text{skip}, s, (h \mid p + 4i \mapsto \llbracket E' \rrbracket_s).$$

Thus, (*) holds.

- When $C' = \text{skip} \wedge s' = s \wedge h' = (h \mid p + 4i \mapsto \llbracket E' \rrbracket_s)$:
Let $\mathbf{s}' = \mathbf{s}$ and $\mathbf{h}' = (\mathbf{h} \mid (\ell, i) \mapsto \llbracket E' \rrbracket_{\mathbf{s}})$.
To prove (**), it suffices to show that
(1) $\mathbf{s}', \mathbf{h}' \models_{\text{dom}(\mathbf{T})} E \hookrightarrow E'$; and
(2) $\mathbf{h}' \uplus \mathbf{h}_{\mathbf{F}} \approx_{\mathbf{T}} h'$.
- $\mathbf{s}', \mathbf{h}' \models_{\text{dom}(\mathbf{T})} E \hookrightarrow E'$ follows from
 - $\llbracket E \rrbracket_{\mathbf{s}'} = \llbracket E \rrbracket_{\mathbf{s}} = \ell \hat{+} 4i$;
 - $\llbracket E' \rrbracket_{\mathbf{s}'} = \llbracket E' \rrbracket_{\mathbf{s}} = \mathbf{h}'(\ell)(i) \neq \text{undef}$.
- We show $\mathbf{h}' \uplus \mathbf{h}_{\mathbf{F}} \approx_{\mathbf{T}} h'$ as follows.
- From $\mathbf{h} \uplus \mathbf{h}_{\mathbf{F}} : \mathbf{T} \wedge \text{Span}(\mathbf{h}') = \text{Span}(\mathbf{h})$, we have
 $\checkmark \mathbf{h}' \uplus \mathbf{h}_{\mathbf{F}} : \mathbf{T}$.
- From $\text{phyh}_{\mathbf{T}}(\mathbf{h} \uplus \mathbf{h}_{\mathbf{F}}) \subseteq h \wedge \llbracket E' \rrbracket_s = \text{phyv}_{\mathbf{T}}(\llbracket E' \rrbracket_{\mathbf{s}})$, we have
 $\checkmark \text{phyh}_{\mathbf{T}}(\mathbf{h}' \uplus \mathbf{h}_{\mathbf{F}}) \subseteq h'$.
- From $\mathbf{h} \uplus \mathbf{h}_{\mathbf{F}} :: \mathbf{T} \wedge \llbracket E' \rrbracket_{\mathbf{s}} \in \text{Safe}(\text{dom}(\mathbf{T}))$, we have
 $\checkmark \mathbf{h}' \uplus \mathbf{h}_{\mathbf{F}} :: \mathbf{T}$.
- Now it suffices to show $\text{shape}(\mathbf{T}) \subseteq I_{\text{gc}}(\text{dom}(\text{shape}(\mathbf{T})), h')$.
- From $\mathbf{h} \uplus \mathbf{h}_{\mathbf{F}} \approx_{\mathbf{T}} h$, we have σ such that
 $\checkmark \sigma = I_{\text{gc}}(\text{dom}(\text{shape}(\mathbf{T})), h) \wedge \text{shape}(\mathbf{T}) \subseteq \sigma$.

- By GCAXiom_2 for $\sigma = I_{\text{gc}}(\text{dom}(\text{shape}(\mathbf{T})), h)$, we have σ' such that
 $\checkmark \sigma' = I_{\text{gc}}(\text{dom}(\text{shape}(\mathbf{T})), h') \wedge \sigma' \subseteq \sigma$
because
 - $\overline{\text{dom}(\sigma)} \subseteq \text{dom}(h) = \text{dom}(h')$ holds by GCAXiom_1 ;
 - $\text{reach}(\text{dom}(\text{shape}(\mathbf{T})), h', \sigma) \subseteq \text{dom}(\text{shape}(\mathbf{T})) \subseteq \text{dom}(\sigma)$ follows, by Lemma 10, from
 $\mathbf{h}' \uplus \mathbf{h}_F \sim_{\mathbf{T}} h' \wedge \mathbf{h}' \uplus \mathbf{h}_F :: \mathbf{T}$ and $\text{shape}(\mathbf{T}) \subseteq \sigma$;
 - $\forall p' \notin \overline{\text{dom}(\sigma)}. h'(p') = h(p')$ holds since $p + 4i \in \overline{\text{dom}(\text{shape}(\mathbf{T}))} \subseteq \overline{\text{dom}(\sigma)}$.
- Now it suffices to show that $\text{shape}(\mathbf{T}) \subseteq \sigma'$, which follows from
 - (1) $\text{shape}(\mathbf{T}) \subseteq \sigma \wedge \sigma' \subseteq \sigma$; and
 - (2) $\text{dom}(\text{shape}(\mathbf{T})) \subseteq \text{reach}(\text{dom}(\text{shape}(\mathbf{T})), h', \sigma') \subseteq \text{dom}(\sigma')$ by GCAXiom_1 .

□

5.2.5 Seq

Lemma 11 (Soundness: Generalized Seq).

$$\frac{\{\mathbf{P}\} C_1 \{\mathbf{Q}\} : k \quad \{\mathbf{Q}\} C_2 \{\mathbf{R}\} : k}{\{\mathbf{P}\} C_1; C_2 \{\mathbf{R}\} : k}$$

Proof.

- Assume: $\{\mathbf{P}\} C_1 \{\mathbf{Q}\} : k$
- Assume: $\{\mathbf{Q}\} C_2 \{\mathbf{R}\} : k$
- Assume: $\rho \in \text{Env}(\text{FLV}(\mathbf{P}, \mathbf{R})), j, \mathbf{s}, \mathbf{h}, \mathbf{h}_F, \mathbf{T}, s, h, C', s', h'$ such that
 $j \leq k \wedge \mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} \mathbf{P}[\rho] \wedge \mathbf{s} \sim_{\mathbf{T}} s \wedge \mathbf{h} \uplus \mathbf{h}_F \approx_{\mathbf{T}} h \wedge (C_1; C_2, s, h \rightsquigarrow^j C', s', h')$
- To show:
 - (*) $(C', s', h' \rightsquigarrow -) \vee$
 - (**) $(\exists \mathbf{s}', \mathbf{h}'. C' = \text{skip} \wedge \mathbf{s}', \mathbf{h}' \models_{\text{dom}(\mathbf{T})} \mathbf{R}[\rho] \wedge$
 $(\forall \mathbf{y} \notin \text{Mod}(C_1; C_2). \mathbf{s}'(\mathbf{y}) = \mathbf{s}(\mathbf{y})) \wedge \mathbf{s}' \sim_{\mathbf{T}} s' \wedge \mathbf{h}' \uplus \mathbf{h}_F \approx_{\mathbf{T}} h')$
- Let $\rho' := \rho|_{\text{FLV}(\mathbf{Q})}$.
- Then, as $\mathbf{P}[\rho] = \mathbf{P}[\rho']$, we have
 $\checkmark \mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} \mathbf{P}[\rho']$.
- From $C_1; C_2, s, h \rightsquigarrow^j C', s', h'$, we have two cases.
- When $C_1, s, h \rightsquigarrow^j C'_1, s', h' \wedge C' = C'_1; C_2$:
 - By assumption we have two cases.
 - When $C'_1, s', h' \rightsquigarrow -$:
(*) holds because $(C'_1; C_2), s', h' \rightsquigarrow -$.

- When $C'_1 = \text{skip} \wedge (s', \mathbf{h}' \models_{\text{dom}(\mathbf{T})} \mathbf{Q}[\rho']) \wedge (\forall y \notin \text{Mod}(C_1). s'(y) = \mathbf{s}(y)) \wedge s' \sim_{\mathbf{T}} s' \wedge \mathbf{h}' \uplus \mathbf{h}_{\mathbf{F}} \approx_{\mathbf{T}} h'$ for some s', \mathbf{h}' :
 $(*)$ holds because $(\text{skip}; C_2), s', h' \rightsquigarrow C_2, s', h'$.
- When $C_1, s, h \rightsquigarrow^{j_1} \text{skip}, s'_1, h'_1 \wedge C_2, s'_1, h'_1 \rightsquigarrow^{j_2} C', s', h' \wedge j = j_1 + j_2 + 1$:
 - As $j_1 \leq k \wedge C_1, s, h \rightsquigarrow^{j_1}$, by assumption, we have s'_1, \mathbf{h}'_1 such that
 $\checkmark s'_1, \mathbf{h}'_1 \models_{\text{dom}(\mathbf{T})} \mathbf{Q}[\rho'] \wedge (\forall y \notin \text{Mod}(C_1). s'_1(y) = \mathbf{s}(y)) \wedge s'_1 \sim_{\mathbf{T}} s'_1 \wedge \mathbf{h}'_1 \uplus \mathbf{h}_{\mathbf{F}} \approx_{\mathbf{T}} h'_1$.
 - As $j_2 \leq k \wedge C_2, s'_1, h'_1 \rightsquigarrow^{j_2} C', s', h'$, by assumption we have two cases.
 - When $C', s', h' \rightsquigarrow -$:
 $(*)$ holds.
 - When $C' = \text{skip} \wedge s', \mathbf{h}' \models_{\text{dom}(\mathbf{T})} \mathbf{R}[\rho'] \wedge (\forall y \notin \text{Mod}(C_2). s'(y) = s'_1(y)) \wedge s' \sim_{\mathbf{T}} s' \wedge \mathbf{h}' \uplus \mathbf{h}_{\mathbf{F}} \approx_{\mathbf{T}} h'$:
 $(**)$ holds because
 - (1) $s', \mathbf{h}' \models_{\text{dom}(\mathbf{T})} \mathbf{R}[\rho]$ holds since $\mathbf{R}[\rho'] = \mathbf{R}[\rho]$;
 - (2) $(\forall y \notin \text{Mod}(C_1; C_2). s'(y) = \mathbf{s}(y))$ follows from $(\forall y \notin \text{Mod}(C_2). s'(y) = s'_1(y))$ and $(\forall y \notin \text{Mod}(C_1). s'_1(y) = \mathbf{s}(y))$ since $\text{Mod}(C_1; C_2) = \text{Mod}(C_1) \cup \text{Mod}(C_2)$.

□

Theorem 5 (Soundness: Seq (partial)).

$$\frac{\{\mathbf{P}\} C_1 \{\mathbf{Q}\} \quad \{\mathbf{Q}\} C_2 \{\mathbf{R}\}}{\{\mathbf{P}\} C_1; C_2 \{\mathbf{R}\}}$$

Proof. It holds by Lemma 11. □

Theorem 6 (Soundness: Seq (total)).

$$\frac{[\mathbf{P}] C_1 [\mathbf{Q}] \quad [\mathbf{Q}] C_2 [\mathbf{R}]}{[\mathbf{P}] C_1; C_2 [\mathbf{R}]}$$

Proof.

- Assume $[\mathbf{P}] C_1 [\mathbf{Q}]$.
- Assume $[\mathbf{Q}] C_2 [\mathbf{R}]$.
- By Theorem 5, we have $\{\mathbf{P}\} C_1; C_2 \{\mathbf{R}\}$.
- Assume: $\rho \in \text{Env}(\text{FLV}(\mathbf{P}, \mathbf{R}))$, $\mathbf{s}, \mathbf{h}, \mathbf{h}_{\mathbf{F}}, \mathbf{T}, s, h$ such that
 $\mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} \mathbf{P}[\rho] \wedge \mathbf{s} \sim_{\mathbf{T}} s \wedge \mathbf{h} \uplus \mathbf{h}_{\mathbf{F}} \approx_{\mathbf{T}} h$.
- Now we show $\neg(C_1; C_2, s, h \text{ diverges})$ by contradiction.
- Assume $\{D_i, s_i, h_i\}_{i \in \mathbb{N}}$ such that
 $\checkmark (D_0, s_0, h_0) = (C_1; C_2, s, h) \wedge \forall i. D_i, s_i, h_i \rightsquigarrow D_{i+1}, s_{i+1}, h_{i+1}$.
- Let $\rho' := \rho|^{\text{FLV}(\mathbf{Q})}$.

- Then, as $\mathbf{P}[\rho] = \mathbf{P}[\rho']$, we have $\mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} \mathbf{P}[\rho']$.
- By $[\mathbf{P}] C_1 [\mathbf{Q}]$, we have $\neg(C_1, s, h \text{ diverges})$.
- Thus, we have some k such that $D_k = (\text{skip}; C_2)$ and $C_1, s, h \rightsquigarrow^k \text{skip}, s_k, h_k$.
- As $D_k = (\text{skip}; C_2)$, we have $D_{k+1} = C_2$, $s_{k+1} = s_k$, and $h_{k+1} = h_k$.
- By $[\mathbf{P}] C_1 [\mathbf{Q}]$, we have $\mathbf{s}_k, \mathbf{h}_k$ such that
 $\checkmark \mathbf{s}_k, \mathbf{h}_k \models_{\text{dom}(\mathbf{T})} \mathbf{Q}[\rho'] \wedge (\forall \mathbf{y} \notin \text{Mod}(C_1). \mathbf{s}_k(\mathbf{y}) = \mathbf{s}(\mathbf{y})) \wedge \mathbf{s}_k \sim_{\mathbf{T}} s_k \wedge \mathbf{h}_k \uplus \mathbf{h}_F \approx_{\mathbf{T}} h_k$.
- By $[\mathbf{Q}] C_2 [\mathbf{R}]$, we have $\neg(C_2, s_k, h_k \text{ diverges})$.
- Thus we have $\neg(D_{k+1}, s_{k+1}, h_{k+1} \text{ diverges})$, which is a contradiction.

□

5.2.6 Frame

Theorem 7 (Soundness: Frame).

$$\frac{\{\mathbf{P}\} C \{\mathbf{Q}\} \quad \text{FPV}(\mathbf{R}) \cap \text{Mod}(C) = \emptyset}{\{\mathbf{P} * \mathbf{R}\} C \{\mathbf{Q} * \mathbf{R}\}} \quad \frac{[\mathbf{P}] C [\mathbf{Q}] \quad \text{FPV}(\mathbf{R}) \cap \text{Mod}(C) = \emptyset}{[\mathbf{P} * \mathbf{R}] C [\mathbf{Q} * \mathbf{R}]}$$

Proof.

- Assume: $\text{FPV}(\mathbf{R}) \cap \text{Mod}(C) = \emptyset$
- Assume: $\forall \rho \in \text{Env}(\text{FLV}(\mathbf{P}, \mathbf{Q})), \mathbf{s}, \mathbf{h}, \mathbf{h}_F, \mathbf{T}, s, h, C', s', h'$.
 $\mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} \mathbf{P}[\rho] \wedge \mathbf{s} \sim_{\mathbf{T}} s \wedge \mathbf{h} \uplus \mathbf{h}_F \approx_{\mathbf{T}} h \wedge C, s, h \rightsquigarrow^* C', s', h' \implies$
 $((C', s', h' \rightsquigarrow -) \vee$
 $(\exists \mathbf{s}', \mathbf{h}'. C' = \text{skip} \wedge \mathbf{s}', \mathbf{h}' \models_{\text{dom}(\mathbf{T})} \mathbf{Q}[\rho] \wedge$
 $(\forall \mathbf{y} \notin \text{Mod}(C). \mathbf{s}'(\mathbf{y}) = \mathbf{s}(\mathbf{y})) \wedge \mathbf{s}' \sim_{\mathbf{T}} s' \wedge \mathbf{h}' \uplus \mathbf{h}_F \approx_{\mathbf{T}} h'))$
 $[\# \wedge \neg(C, s, h \text{ diverges}) \#]$
- Assume: $\rho \in \text{Env}(\text{FLV}(\mathbf{P}, \mathbf{Q}, \mathbf{R})), \mathbf{s}, \mathbf{h}, \mathbf{h}_F, \mathbf{T}, s, h, C', s', h'$ such that
 $\mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} (\mathbf{P}[\rho] * \mathbf{R}[\rho]) \wedge \mathbf{s} \sim_{\mathbf{T}} s \wedge \mathbf{h} \uplus \mathbf{h}_F \approx_{\mathbf{T}} h \wedge C, s, h \rightsquigarrow^* C', s', h'$
- To show:
 $[\# \neg(C, s, h \text{ diverges}); \text{ and } \#]$
 $(*) (C', s', h' \rightsquigarrow -) \vee$
 $(**) (\exists \mathbf{s}', \mathbf{h}'. C' = \text{skip} \wedge \mathbf{s}', \mathbf{h}' \models_{\text{dom}(\mathbf{T})} (\mathbf{Q}[\rho] * \mathbf{R}[\rho]) \wedge$
 $(\forall \mathbf{y} \notin \text{Mod}(C). \mathbf{s}'(\mathbf{y}) = \mathbf{s}(\mathbf{y})) \wedge \mathbf{s}' \sim_{\mathbf{T}} s' \wedge \mathbf{h}' \uplus \mathbf{h}_F \approx_{\mathbf{T}} h')$
- From $\mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} (\mathbf{P}[\rho] * \mathbf{R}[\rho])$, we have \mathbf{h}_1 and \mathbf{h}_2 such that
 $\checkmark \mathbf{h} = \mathbf{h}_1 \uplus \mathbf{h}_2,$
 $\checkmark \mathbf{s}, \mathbf{h}_1 \models_{\text{dom}(\mathbf{T})} \mathbf{P}[\rho],$
 $\checkmark \mathbf{s}, \mathbf{h}_2 \models_{\text{dom}(\mathbf{T})} \mathbf{R}[\rho].$
- $[\# \neg(C, s, h \text{ diverges})$ holds by assumption since $\mathbf{h} \uplus \mathbf{h}_F = \mathbf{h}_1 \uplus (\mathbf{h}_2 \uplus \mathbf{h}_F) \wedge \mathbf{s}, \mathbf{h}_1 \models_{\text{dom}(\mathbf{T})} \mathbf{P}[\rho] \#]$
- Also by assumption we have two cases since $\mathbf{h} \uplus \mathbf{h}_F = \mathbf{h}_1 \uplus (\mathbf{h}_2 \uplus \mathbf{h}_F) \wedge \mathbf{s}, \mathbf{h}_1 \models_{\text{dom}(\mathbf{T})} \mathbf{P}[\rho].$

- When $C', s', h' \rightsquigarrow -$:
(*) holds.
- When $C' = \text{skip} \wedge (s', \mathbf{h}' \models_{\text{dom}(\mathbf{T})} \mathbf{Q}[\rho]) \wedge (\forall y \notin \text{Mod}(C). s'(y) = \mathbf{s}(y)) \wedge s' \sim_{\mathbf{T}} s' \wedge \mathbf{h}' \uplus \mathbf{h}_2 \uplus \mathbf{h}_{\mathbf{F}} \approx_{\mathbf{T}} h'$ for some s', \mathbf{h}' :
(**) is shown as follows.
- To show (**), it suffices to show that $s', \mathbf{h}' \uplus \mathbf{h}_2 \models_{\text{dom}(\mathbf{T})} \mathbf{Q}[\rho] * \mathbf{R}[\rho]$.
- We split the heap $\mathbf{h}' \uplus \mathbf{h}_2$ into \mathbf{h}' and \mathbf{h}_2 .
- As $s', \mathbf{h}' \models_{\text{dom}(\mathbf{T})} \mathbf{Q}[\rho]$ holds, we need to show $s', \mathbf{h}_2 \models_{\text{dom}(\mathbf{T})} \mathbf{R}[\rho]$, which follows from $(\mathbf{s}, \mathbf{h}_2 \models_{\text{dom}(\mathbf{T})} \mathbf{R}[\rho]) \wedge (\forall y \notin \text{Mod}(C). \mathbf{s}(y) = \mathbf{s}(y)) \wedge \text{FPV}(\mathbf{R}) \cap \text{Mod}(C) = \emptyset$ by Lemma 5.

□

5.2.7 Conseq

Theorem 8 (Soundness: Conseq).

$$\frac{\mathbf{P} \models \mathbf{P}' \quad \{ \mathbf{P}' \} C \{ \mathbf{Q}' \} \quad \mathbf{Q}' \models \mathbf{Q}}{\{ \mathbf{P} \} C \{ \mathbf{Q} \}} \quad \frac{\mathbf{P} \models \mathbf{P}' \quad [\mathbf{P}'] C [\mathbf{Q}'] \quad \mathbf{Q}' \models \mathbf{Q}}{[\mathbf{P}] C [\mathbf{Q}]}$$

Proof.

- Assume: $\mathbf{P} \models \mathbf{P}'$ and $\mathbf{Q}' \models \mathbf{Q}$.
- Assume: $\forall \rho \in \text{Env}(\text{FLV}(\mathbf{P}', \mathbf{Q}')), \mathbf{s}, \mathbf{h}, \mathbf{h}_{\mathbf{F}}, \mathbf{T}, s, h, C', s', h'$.
 $\mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} \mathbf{P}'[\rho] \wedge \mathbf{s} \sim_{\mathbf{T}} s \wedge \mathbf{h} \uplus \mathbf{h}_{\mathbf{F}} \approx_{\mathbf{T}} h \wedge C, s, h \rightsquigarrow^* C', s', h' \implies$
 $((C', s', h' \rightsquigarrow -) \vee$
 $(\exists s', \mathbf{h}'. C' = \text{skip} \wedge s', \mathbf{h}' \models_{\text{dom}(\mathbf{T})} \mathbf{Q}'[\rho] \wedge$
 $(\forall y \notin \text{Mod}(C). s'(y) = \mathbf{s}(y)) \wedge s' \sim_{\mathbf{T}} s' \wedge \mathbf{h}' \uplus \mathbf{h}_{\mathbf{F}} \approx_{\mathbf{T}} h'))$
 $[\# \wedge \neg(C, s, h \text{ diverges}) \#]$
- Assume: $\rho \in \text{Env}(\text{FLV}(\mathbf{P}, \mathbf{Q})), \mathbf{s}, \mathbf{h}, \mathbf{h}_{\mathbf{F}}, \mathbf{T}, s, h, C', s', h'$ such that
 $\mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} \mathbf{P}[\rho] \wedge \mathbf{s} \sim_{\mathbf{T}} s \wedge \mathbf{h} \uplus \mathbf{h}_{\mathbf{F}} \approx_{\mathbf{T}} h \wedge C, s, h \rightsquigarrow^* C', s', h'$
- To show:
 $[\# \neg(C, s, h \text{ diverges})];$ and $\#]$
(*) $(C', s', h' \rightsquigarrow -) \vee$
(**) $(\exists s', \mathbf{h}'. C' = \text{skip} \wedge s', \mathbf{h}' \models_{\text{dom}(\mathbf{T})} \mathbf{Q}'[\rho] \wedge$
 $(\forall y \notin \text{Mod}(C). s'(y) = \mathbf{s}(y)) \wedge s' \sim_{\mathbf{T}} s' \wedge \mathbf{h}' \uplus \mathbf{h}_{\mathbf{F}} \approx_{\mathbf{T}} h')$
- Let $\rho' := \rho|_{\text{FLV}(\mathbf{P}', \mathbf{Q}')}.$
- From $\mathbf{P} \models \mathbf{P}'$ and $\mathbf{s} \sim_{\mathbf{T}} s \wedge \mathbf{h} \uplus \mathbf{h}_{\mathbf{F}} \approx_{\mathbf{T}} h \wedge \mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} \mathbf{P}[\rho']$ (as $\mathbf{P}[\rho'] = \mathbf{P}[\rho]$), we have
 $\checkmark \mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} \mathbf{P}'[\rho'].$
- $[\# \neg(C, s, h \text{ diverges})$ holds by assumption. $\#]$
- Also by assumption we have two cases.

- When $C', s', h' \rightsquigarrow -$:
(*) holds.
- When $C' = \text{skip} \wedge (s', \mathbf{h}' \models_{\text{dom}(\mathbf{T})} \mathbf{Q}'[\rho']) \wedge (\forall y \notin \text{Mod}(C). s'(y) = \mathbf{s}(y)) \wedge s' \sim_{\mathbf{T}} s' \wedge \mathbf{h}' \uplus \mathbf{h}_{\mathbf{F}} \approx_{\mathbf{T}} h'$ for some s', \mathbf{h}' :
(**) holds because $s', \mathbf{h}' \models_{\text{dom}(\mathbf{T})} \mathbf{Q}[\rho]$ follows from $\mathbf{Q}' \models \mathbf{Q}$ and $s' \sim_{\mathbf{T}} s' \wedge \mathbf{h}' \uplus \mathbf{h}_{\mathbf{F}} \approx_{\mathbf{T}} h' \wedge s', \mathbf{h}' \models_{\text{dom}(\mathbf{T})} \mathbf{Q}'[\rho']$ (as $\mathbf{Q}[\rho'] = \mathbf{Q}[\rho]$).

□

5.2.8 Ex

Theorem 9 (Soundness: Ex).

$$\frac{\{\mathbf{P}\} C \{\mathbf{Q}\}}{\{\exists v. \mathbf{P}\} C \{\exists v. \mathbf{Q}\}} \quad \frac{[\mathbf{P}] C [\mathbf{Q}]}{[\exists v. \mathbf{P}] C [\exists v. \mathbf{Q}]}$$

Proof.

- Assume: $\forall \rho \in \text{Env}(\text{FLV}(\mathbf{P}, \mathbf{Q})), \mathbf{s}, \mathbf{h}, \mathbf{h}_{\mathbf{F}}, \mathbf{T}, s, h, C', s', h'$.
 $\mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} \mathbf{P}[\rho] \wedge s \sim_{\mathbf{T}} s \wedge \mathbf{h} \uplus \mathbf{h}_{\mathbf{F}} \approx_{\mathbf{T}} h \wedge C, s, h \rightsquigarrow^* C', s', h' \implies$
 $((C', s', h' \rightsquigarrow -) \vee$
 $(\exists s', \mathbf{h}'. C' = \text{skip} \wedge s', \mathbf{h}' \models_{\text{dom}(\mathbf{T})} \mathbf{Q}[\rho] \wedge$
 $(\forall y \notin \text{Mod}(C). s'(y) = \mathbf{s}(y)) \wedge s' \sim_{\mathbf{T}} s' \wedge \mathbf{h}' \uplus \mathbf{h}_{\mathbf{F}} \approx_{\mathbf{T}} h'))$
 $[\# \wedge \neg(C, s, h \text{ diverges}) \#]$
- Assume: $\rho \in \text{Env}(\text{FLV}(\exists v. \mathbf{P}, \exists v. \mathbf{Q})), \mathbf{s}, \mathbf{h}, \mathbf{h}_{\mathbf{F}}, \mathbf{T}, s, h, C', s', h'$ such that
 $(\mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} (\exists v. \mathbf{P})[\rho]) \wedge s \sim_{\mathbf{T}} s \wedge \mathbf{h} \uplus \mathbf{h}_{\mathbf{F}} \approx_{\mathbf{T}} h \wedge C, s, h \rightsquigarrow^* C', s', h'$
- To show:
 $[\# \neg(C, s, h \text{ diverges})];$ and $\#]$
(*) $(C', s', h' \rightsquigarrow -) \vee$
(**) $(\exists s', \mathbf{h}'. C' = \text{skip} \wedge s', \mathbf{h}' \models_{\text{dom}(\mathbf{T})} (\exists v. \mathbf{Q})[\rho] \wedge$
 $(\forall y \notin \text{Mod}(C). s'(y) = \mathbf{s}(y)) \wedge s' \sim_{\mathbf{T}} s' \wedge \mathbf{h}' \uplus \mathbf{h}_{\mathbf{F}} \approx_{\mathbf{T}} h')$
- From $\mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} (\exists v. \mathbf{P})[\rho]$, by Lemma 7 we have
 $\checkmark \mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} \mathbf{P}[(\rho \mid v \mapsto \mathbf{v})]$ for some $\mathbf{v} \in \text{LogVals}$.
- Let $\rho' := (\rho \mid v \mapsto \mathbf{v})$.
- $[\# \neg(C, s, h \text{ diverges})$ holds by assumption. $\#]$
- Also by assumption we have two cases.
- When $C', s', h' \rightsquigarrow -$:
(*) holds.
- When $C' = \text{skip} \wedge (s', \mathbf{h}' \models_{\text{dom}(\mathbf{T})} \mathbf{Q}[\rho']) \wedge (\forall y \notin \text{Mod}(C). s'(y) = \mathbf{s}(y)) \wedge s' \sim_{\mathbf{T}} s' \wedge \mathbf{h}' \uplus \mathbf{h}_{\mathbf{F}} \approx_{\mathbf{T}} h'$ for some s', \mathbf{h}' :
(**) holds because $s', \mathbf{h}' \models_{\text{dom}(\mathbf{T})} (\exists v. \mathbf{Q})[\rho]$ follows from $s', \mathbf{h}' \models_{\text{dom}(\mathbf{T})} \mathbf{Q}[\rho']$ by Lemma 7.

□

5.2.9 Gen

Theorem 10 (Soundness: Gen).

$$\frac{\forall v \in \text{LogVals. } \{\mathbf{P}[v/v]\} C \{\mathbf{Q}[v/v]\}}{\{\mathbf{P}\} C \{\mathbf{Q}\}} \quad \frac{\forall v \in \text{LogVals. } [\mathbf{P}[v/v]] C [\mathbf{Q}[v/v]]}{[\mathbf{P}] C [\mathbf{Q}]}$$

Proof. The goal directly follows by definition because $\mathbf{P}[\rho] = \mathbf{P}[\rho(v)/v][\rho]$ and $\mathbf{Q}[\rho] = \mathbf{Q}[\rho(v)/v][\rho]$ for any $\rho \in \text{Env}(\text{FLV}(\mathbf{P}, \mathbf{Q}))$. \square

5.2.10 Total

Theorem 11 (Soundness: Total).

$$\frac{[\mathbf{P}] C [\mathbf{Q}]}{\{\mathbf{P}\} C \{\mathbf{Q}\}}$$

Proof. It holds vacuously by definition. \square

5.2.11 If

Theorem 12 (Soundness: If).

$$\frac{\{\mathbf{P} \wedge E\} C_1 \{\mathbf{Q}\} \quad \{\mathbf{P} \wedge \text{not } E\} C_2 \{\mathbf{Q}\}}{\{\mathbf{P} \wedge \text{word}(E)\} \text{ if } E \text{ then } C_1 \text{ else } C_2 \text{ fi } \{\mathbf{Q}\}} \quad \frac{[\mathbf{P} \wedge E] C_1 [\mathbf{Q}] \quad [\mathbf{P} \wedge \text{not } E] C_2 [\mathbf{Q}]}{[\mathbf{P} \wedge \text{word}(E)] \text{ if } E \text{ then } C_1 \text{ else } C_2 \text{ fi } [\mathbf{Q}]}$$

Proof.

- Assume: $\forall \rho \in \text{Env}(\text{FLV}(\mathbf{P}, \mathbf{Q})), \mathbf{s}, \mathbf{h}, \mathbf{h}_F, \mathbf{T}, s, h, C', s', h'$.
 $(\mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} \mathbf{P}[\rho] \wedge E) \wedge \mathbf{s} \sim_{\mathbf{T}} s \wedge \mathbf{h} \uplus \mathbf{h}_F \approx_{\mathbf{T}} h \wedge C_1, s, h \rightsquigarrow^* C', s', h' \implies$
 $((C', s', h' \rightsquigarrow -) \vee$
 $(\exists \mathbf{s}', \mathbf{h}'. C' = \text{skip} \wedge \mathbf{s}', \mathbf{h}' \models_{\text{dom}(\mathbf{T})} \mathbf{Q}[\rho] \wedge$
 $(\forall y \notin \text{Mod}(C_1). \mathbf{s}'(y) = \mathbf{s}(y)) \wedge \mathbf{s}' \sim_{\mathbf{T}} s' \wedge \mathbf{h}' \uplus \mathbf{h}_F \approx_{\mathbf{T}} h'))$
 $[\# \wedge \neg(C_1, s, h \text{ diverges}) \#]$
- Assume: $\forall \rho \in \text{Env}(\text{FLV}(\mathbf{P}, \mathbf{Q})), \mathbf{s}, \mathbf{h}, \mathbf{h}_F, \mathbf{T}, s, h, C', s', h'$.
 $(\mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} \mathbf{P}[\rho] \wedge \text{not } E) \wedge \mathbf{s} \sim_{\mathbf{T}} s \wedge \mathbf{h} \uplus \mathbf{h}_F \approx_{\mathbf{T}} h \wedge C_2, s, h \rightsquigarrow^* C', s', h' \implies$
 $((C', s', h' \rightsquigarrow -) \vee$
 $(\exists \mathbf{s}', \mathbf{h}'. C' = \text{skip} \wedge \mathbf{s}', \mathbf{h}' \models_{\text{dom}(\mathbf{T})} \mathbf{Q}[\rho] \wedge$
 $(\forall y \notin \text{Mod}(C_2). \mathbf{s}'(y) = \mathbf{s}(y)) \wedge \mathbf{s}' \sim_{\mathbf{T}} s' \wedge \mathbf{h}' \uplus \mathbf{h}_F \approx_{\mathbf{T}} h'))$
 $[\# \wedge \neg(C_2, s, h \text{ diverges}) \#]$
- Assume: $\rho \in \text{Env}(\text{FLV}(\mathbf{P}, \mathbf{Q})), \mathbf{s}, \mathbf{h}, \mathbf{h}_F, \mathbf{T}, s, h, C', s', h'$ such that
 $(\mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} \mathbf{P}[\rho] \wedge \text{word}(E)) \wedge \mathbf{s} \sim_{\mathbf{T}} s \wedge \mathbf{h} \uplus \mathbf{h}_F \approx_{\mathbf{T}} h \wedge \text{if } E \text{ then } C_1 \text{ else } C_2 \text{ fi}, s, h \rightsquigarrow^* C', s', h'$
- To show:
 $[\# \neg(\text{if } E \text{ then } C_1 \text{ else } C_2 \text{ fi}, s, h \text{ diverges}); \text{ and } \#]$
 $(*) (C', s', h' \rightsquigarrow -) \vee$
 $(**) (\exists \mathbf{s}', \mathbf{h}'. C' = \text{skip} \wedge \mathbf{s}', \mathbf{h}' \models_{\text{dom}(\mathbf{T})} \mathbf{Q}[\rho] \wedge$
 $(\forall y \notin \text{Mod}(C_1, C_2). \mathbf{s}'(y) = \mathbf{s}(y)) \wedge \mathbf{s}' \sim_{\mathbf{T}} s' \wedge \mathbf{h}' \uplus \mathbf{h}_F \approx_{\mathbf{T}} h')$

- From $\mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} \text{word}(E)$, we have
 $\checkmark \llbracket E \rrbracket_{\mathbf{s}} \in \text{Words}$.
- By Lemma 2, we have
 $\checkmark \llbracket E \rrbracket_{\mathbf{s}} = \text{phyv}_{\mathbf{T}}(\llbracket E \rrbracket_{\mathbf{s}}) = \llbracket E \rrbracket_{\mathbf{s}}$.
- Thus, we have two cases.
- When $\llbracket E \rrbracket_{\mathbf{s}} \in \text{Words} \setminus \{0\}$:
 - [$\#$ Since we have if E then C_1 else C_2 fi, $s, h \rightsquigarrow C_1, s, h$ and $\mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} \mathbf{P}[\rho] \wedge E$, by assumption we have $\neg(C_1, s, h \text{ diverges})$ and thus $\neg(C, s, h \text{ diverges})$ holds. $\#$]
 - From if E then C_1 else C_2 fi, $s, h \rightsquigarrow^* C', s', h'$ we have two cases.
 - When $C' = \text{if } E \text{ then } C_1 \text{ else } C_2 \text{ fi} \wedge s' = s \wedge h' = h$:
 $(*)$ holds as we have if E then C_1 else C_2 fi, $s, h \rightsquigarrow C_1, s, h$.
 - When $C_1, s, h \rightsquigarrow^* C', s', h'$:
 $(*)$ or $(**)$ holds by assumption since we have $\mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} \mathbf{P}[\rho] \wedge E$.
- When $\llbracket E \rrbracket_{\mathbf{s}} = 0$:
 - [$\#$ Since we have if E then C_1 else C_2 fi, $s, h \rightsquigarrow C_2, s, h$ and $\mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} \mathbf{P}[\rho] \wedge \text{not } E$, by assumption we have $\neg(C_2, s, h \text{ diverges})$ and thus $\neg(C, s, h \text{ diverges})$ holds. $\#$]
 - From if E then C_1 else C_2 fi, $s, h \rightsquigarrow^* C', s', h'$ we have two cases.
 - When $C' = \text{if } E \text{ then } C_1 \text{ else } C_2 \text{ fi} \wedge s' = s \wedge h' = h$:
 $(*)$ holds as we have if E then C_1 else C_2 fi, $s, h \rightsquigarrow C_2, s, h$.
 - When $C_2, s, h \rightsquigarrow^* C', s', h'$:
 $(*)$ or $(**)$ holds by assumption since we have $\mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} \mathbf{P}[\rho] \wedge \text{not } E$.

□

5.2.12 While

Theorem 13 (Soundness: While).

$$\frac{\{\mathbf{P} \wedge E\} C \{\mathbf{P} \wedge \text{word}(E)\}}{\{\mathbf{P} \wedge \text{word}(E)\} \text{ while } E \text{ do } C \text{ od } \{\mathbf{P} \wedge \text{not } E\}}$$

Proof.

- Assume: $\{\mathbf{P} \wedge E\} C \{\mathbf{P} \wedge \text{word}(E)\}$
- To show: $\forall k. \{\mathbf{P} \wedge \text{word}(E)\} \text{ while } E \text{ do } C \text{ od } \{\mathbf{P} \wedge \text{not } E\} : k$
- We prove the goal by induction on k .
- (Base case) when $k = 0$,
 - Assume: $\rho \in \text{Env}(\text{FLV}(\mathbf{P})), \mathbf{s}, \mathbf{h}, \mathbf{h}_F, \mathbf{T}, s, h, C', s', h'$ such that
 $(\mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} \mathbf{P}[\rho] \wedge \text{word}(E)) \wedge \mathbf{s} \sim_{\mathbf{T}} s \wedge \mathbf{h} \uplus \mathbf{h}_F \approx_{\mathbf{T}} h \wedge \text{while } E \text{ do } C \text{ od}, s, h \rightsquigarrow^k C', s', h'$.

- It suffices to show
 - (*) $C', s', h' \rightsquigarrow -$.
- From $\mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} \text{word}(E)$, we have
 - $\checkmark \llbracket E \rrbracket_{\mathbf{s}} \in \text{Words}$.
- By Lemma 2, we have
 - $\checkmark \llbracket E \rrbracket_{\mathbf{s}} = \text{phyv}_{\mathbf{T}}(\llbracket E \rrbracket_{\mathbf{s}}) = \llbracket E \rrbracket_{\mathbf{s}}$.
- (*) holds because $C' = \text{while } E \text{ do } C \text{ od} \wedge s' = s \wedge h' = h$ and $\llbracket E \rrbracket_{\mathbf{s}} \neq \text{undef}$.
- (Inductive step) when $k > 0 \wedge \forall j < k. \{\mathbf{P} \wedge \text{word}(E)\} \text{ while } E \text{ do } C \text{ od} \{\mathbf{P} \wedge \text{not } E\} : j$,
 - Assume: $\rho \in \text{Env}(\text{FLV}(\mathbf{P})), \mathbf{s}, \mathbf{h}, \mathbf{h}_{\mathbf{F}}, \mathbf{T}, s, h, C', s', h'$ such that
 - $(\mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} \mathbf{P}[\rho] \wedge \text{word}(E)) \wedge \mathbf{s} \sim_{\mathbf{T}} s \wedge \mathbf{h} \uplus \mathbf{h}_{\mathbf{F}} \approx_{\mathbf{T}} h \wedge \text{while } E \text{ do } C \text{ od}, s, h \rightsquigarrow^k C', s', h'$.
 - To show:
 - (*) $(C', s', h' \rightsquigarrow -) \vee$
 - (**) $(\exists \mathbf{s}', \mathbf{h}'. C' = \text{skip} \wedge (\mathbf{s}', \mathbf{h}' \models_{\text{dom}(\mathbf{T})} \mathbf{P}[\rho] \wedge \text{not } E) \wedge (\forall \mathbf{y} \notin \text{Mod}(C). \mathbf{s}'(\mathbf{y}) = \mathbf{s}(\mathbf{y})) \wedge \mathbf{s}' \sim_{\mathbf{T}} s' \wedge \mathbf{h}' \uplus \mathbf{h}_{\mathbf{F}} \approx_{\mathbf{T}} h')$
 - From $\mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} \text{word}(E)$, we have
 - $\checkmark \llbracket E \rrbracket_{\mathbf{s}} \in \text{Words}$.
 - By Lemma 2, we have
 - $\checkmark \llbracket E \rrbracket_{\mathbf{s}} = \text{phyv}_{\mathbf{T}}(\llbracket E \rrbracket_{\mathbf{s}}) = \llbracket E \rrbracket_{\mathbf{s}}$.
 - Thus we have two cases.
 - When $\llbracket E \rrbracket_{\mathbf{s}} = \llbracket E \rrbracket_{\mathbf{s}} = 0$:
 - ◊ We have $\text{while } E \text{ do } C \text{ od}, s, h \rightsquigarrow \text{skip}, s, h$.
 - ◊ Thus we have $\text{skip}, s, h \rightsquigarrow^{k-1} C', s', h'$, from which it follows that
 - $\checkmark C' = \text{skip} \wedge s' = s \wedge h' = h$.
 - ◊ Thus (**) holds because we have $\mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} \text{not } E$ from $\llbracket \text{not } E \rrbracket_{\mathbf{s}} = 1$.
 - When $\llbracket E \rrbracket_{\mathbf{s}} = \llbracket E \rrbracket_{\mathbf{s}} \in \text{Words} \setminus \{0\}$:
 - ◊ We have $\text{while } E \text{ do } C \text{ od}, s, h \rightsquigarrow (C; \text{while } E \text{ do } C \text{ od}), s, h$, from which we have
 - $\checkmark (C; \text{while } E \text{ do } C \text{ od}), s, h \rightsquigarrow^{k-1} C', s', h'$.
 - ◊ From $\{\mathbf{P} \wedge E\} C \{\mathbf{P} \wedge \text{word}(E)\}$ and $\{\mathbf{P} \wedge \text{word}(E)\} \text{ while } E \text{ do } C \text{ od} \{\mathbf{P} \wedge \text{not } E\} : k-1$, by Lemma 11 we have
 - $\checkmark \{\mathbf{P} \wedge E\} C; \text{while } E \text{ do } C \text{ od} \{\mathbf{P} \wedge \text{not } E\} : k-1$.
 - ◊ Thus (*) \vee (**) holds since we have $\mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} E$ from $\llbracket E \rrbracket_{\mathbf{s}} \in \text{Words} \setminus \{0\}$.

□

Theorem 14 (Soundness: WhileT).

$$\frac{[\mathbf{P} \wedge E \wedge 0 < \mathbf{E}' = v] C [\mathbf{P} \wedge \text{word}(E) \wedge 0 < \mathbf{E}' < v] \quad v \notin \text{FLV}(\mathbf{P}, \mathbf{E}')}{[\mathbf{P} \wedge \text{word}(E) \wedge 0 < \mathbf{E}'] \text{ while } E \text{ do } C \text{ od} [\mathbf{P} \wedge \text{not } E]}$$

Proof.

- Assume: $[\mathbf{P} \wedge E \wedge 0 < \mathbf{E}' = v] C [\mathbf{P} \wedge \text{word}(E) \wedge 0 < \mathbf{E}' < v]$ and $v \notin \text{FLV}(\mathbf{P}, \mathbf{E}')$.

- By Theorems 8, 9, 11 and 13, we have

$$\begin{array}{c}
\frac{\frac{\frac{\mathbf{P} \wedge E \wedge 0 < \mathbf{E}' = v \quad C \quad [\mathbf{P} \wedge \text{word}(E) \wedge 0 < \mathbf{E}' < v]}{[\exists v. \mathbf{P} \wedge E \wedge 0 < \mathbf{E}' = v] \quad C \quad [\exists v. \mathbf{P} \wedge \text{word}(E) \wedge 0 < \mathbf{E}' < v]} \text{ (Ex)}}{[\mathbf{P} \wedge 0 < \mathbf{E}' \wedge E] \quad C \quad [\mathbf{P} \wedge 0 < \mathbf{E}' \wedge \text{word}(E)]} \text{ (Conseq)}}{\frac{[\mathbf{P} \wedge 0 < \mathbf{E}' \wedge E] \quad C \quad \{\mathbf{P} \wedge 0 < \mathbf{E}' \wedge \text{word}(E)\}}{\{\mathbf{P} \wedge 0 < \mathbf{E}' \wedge \text{word}(E)\} \text{ while } E \text{ do } C \text{ od } \{\mathbf{P} \wedge 0 < \mathbf{E}' \wedge \text{not } E\}} \text{ (Total)}}{\{\mathbf{P} \wedge 0 < \mathbf{E}' \wedge \text{word}(E)\} \text{ while } E \text{ do } C \text{ od } \{\mathbf{P} \wedge 0 < \mathbf{E}' \wedge \text{not } E\}} \text{ (While)}} \\
\frac{\{\mathbf{P} \wedge 0 < \mathbf{E}' \wedge \text{word}(E)\} \text{ while } E \text{ do } C \text{ od } \{\mathbf{P} \wedge 0 < \mathbf{E}' \wedge \text{not } E\}}{\{\mathbf{P} \wedge \text{word}(E) \wedge 0 < \mathbf{E}'\} \text{ while } E \text{ do } C \text{ od } \{\mathbf{P} \wedge \text{not } E\}} \text{ (Conseq)}
\end{array}$$

- Assume: $\rho \in \text{Env}(\text{FLV}(\mathbf{P}, \mathbf{E}'))$, $\mathbf{s}, \mathbf{h}, \mathbf{h}_F, \mathbf{T}, s, h$ such that
 $(\mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} \mathbf{P}[\rho] \wedge \text{word}(E) \wedge 0 < \mathbf{E}'[\rho]) \wedge \mathbf{s} \sim_{\mathbf{T}} s \wedge \mathbf{h} \uplus \mathbf{h}_F \approx_{\mathbf{T}} h$.
- Now we show $\neg(\text{while } E \text{ do } C \text{ od}, s, h \text{ diverges})$ by contradiction.
- Assume: $\{D_i, s_i, h_i\}_{i \in \mathbb{N}}$ such that
 $\checkmark (D_0, s_0, h_0) = (\text{while } E \text{ do } C \text{ od}, s, h) \wedge \forall i. D_i, s_i, h_i \rightsquigarrow D_{i+1}, s_{i+1}, h_{i+1}$.
- We show the following, which is a contradiction because $n_0 > n_1 > n_2 \dots > 0$ is not possible.
- By induction on i , we find $\{k_i, n_i, \mathbf{s}_i, \mathbf{h}_i\}_{i \in \mathbb{N}}$ (with $n_i \in \text{Words}$) such that
 $\checkmark D_{k_i} = \text{while } E \text{ do } C \text{ od};$
 $\checkmark (\mathbf{s}_i, \mathbf{h}_i \models_{\text{dom}(\mathbf{T})} \mathbf{P}[\rho] \wedge \text{word}(E) \wedge 0 < \mathbf{E}'[\rho] = n_i) \wedge \mathbf{s}_i \sim_{\mathbf{T}} s_{k_i} \wedge \mathbf{h}_i \uplus \mathbf{h}_F \approx_{\mathbf{T}} h_{k_i};$
 $\checkmark \text{if } i > 0 \text{ then } 0 < n_i < n_{i-1}$.

(Base Case)

- From $(\mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} 0 < \mathbf{E}'[\rho])$, we have
 $\checkmark \llbracket \mathbf{E}'[\rho] \rrbracket_{\mathbf{s}} \in \text{Words}$.
- Let $k_0 = 0$, $\mathbf{s}_0 = \mathbf{s}$, $\mathbf{h}_0 = \mathbf{h}$ and $n_0 = \llbracket \mathbf{E}'[\rho] \rrbracket_{\mathbf{s}_0} \in \text{Words}$.
- Then by assumption we have
 $\checkmark D_{k_0} = \text{while } E \text{ do } C \text{ od},$
 $\checkmark (\mathbf{s}_0, \mathbf{h}_0 \models_{\text{dom}(\mathbf{T})} \mathbf{P}[\rho] \wedge \text{word}(E) \wedge 0 < \mathbf{E}'[\rho] = n_0) \wedge \mathbf{s}_0 \sim_{\mathbf{T}} s_{k_0} \wedge \mathbf{h}_0 \uplus \mathbf{h}_F \approx_{\mathbf{T}} h_{k_0}$.

(Inductive step)

- Assume:
 $\checkmark D_{k_i} = \text{while } E \text{ do } C \text{ od},$
 $\checkmark (\mathbf{s}_i, \mathbf{h}_i \models_{\text{dom}(\mathbf{T})} \mathbf{P}[\rho] \wedge \text{word}(E) \wedge 0 < \mathbf{E}'[\rho] = n_i) \wedge \mathbf{s}_i \sim_{\mathbf{T}} s_{k_i} \wedge \mathbf{h}_i \uplus \mathbf{h}_F \approx_{\mathbf{T}} h_{k_i}$.
- As $(D_{k_i}, s_{k_i}, h_{k_i})$ diverges, we have
 $\checkmark \llbracket E \rrbracket_{s_{k_i}} \in \text{Words} \setminus \{0\},$
 $\checkmark (D_{k_{i+1}}, s_{k_{i+1}}, h_{k_{i+1}}) = (C; \text{while } E \text{ do } C \text{ od}, s_{k_i}, h_{k_i})$.
- From $\mathbf{s}_i, \mathbf{h}_i \models_{\text{dom}(\mathbf{T})} \text{word}(E)$, we have
 $\checkmark \llbracket E \rrbracket_{\mathbf{s}_i} \in \text{Words}$.
- By Lemma 2, we have $\llbracket E \rrbracket_{\mathbf{s}_i} = \text{phyv}_{\mathbf{T}}(\llbracket E \rrbracket_{\mathbf{s}_i}) = \llbracket E \rrbracket_{s_{k_i}} \in \text{Words} \setminus \{0\}$, and thus we have
 $\checkmark \mathbf{s}_i, \mathbf{h}_i \models_{\text{dom}(\mathbf{T})} E$.

- By $[\mathbf{P} \wedge E \wedge 0 < \mathbf{E}' = v] C [\mathbf{P} \wedge \text{word}(E) \wedge 0 < \mathbf{E}' < v]$, we have
 $\checkmark \neg(C, s_{k_i+1}, h_{k_i+1} \text{ diverges})$.
- Thus, we have some j such that
 $\checkmark D_{k_i+j+1} = (\text{skip}; \text{while } E \text{ do } C \text{ od})$,
 $\checkmark C, s_{k_i+1}, h_{k_i+1} \rightsquigarrow^j \text{skip}, s_{k_i+j+1}, h_{k_i+j+1}$.
- Then, by $[\mathbf{P} \wedge E \wedge 0 < \mathbf{E}' = v] C [\mathbf{P} \wedge \text{word}(E) \wedge 0 < \mathbf{E}' < v]$, we have $\mathbf{s}_{i+1}, \mathbf{h}_{i+1}$ such that
 $\checkmark (\mathbf{s}_{i+1}, \mathbf{h}_{i+1} \models_{\text{dom}(\mathbf{T})} \mathbf{P}[\rho] \wedge \text{word}(E) \wedge 0 < \mathbf{E}'[\rho] < n_i) \wedge \mathbf{s}_{i+1} \sim_{\mathbf{T}} s_{k_i+j+1} \wedge \mathbf{h}_{i+1} \uplus \mathbf{h}_{\mathbf{F}} \approx_{\mathbf{T}} h_{k_i+j+1}$.
- Also we have
 $\checkmark (D_{k_i+j+2}, s_{k_i+j+2}, h_{k_i+j+2}) = (\text{while } E \text{ do } C \text{ od}, s_{k_i+j+1}, h_{k_i+j+1})$.
- From $\mathbf{s}_{i+1}, \mathbf{h}_{i+1} \models_{\text{dom}(\mathbf{T})} 0 < \mathbf{E}'[\rho] < n_i$, we have
 $\checkmark \llbracket \mathbf{E}'[\rho] \rrbracket_{\mathbf{s}_{i+1}} \in \text{Words} \wedge 0 < \llbracket \mathbf{E}'[\rho] \rrbracket_{\mathbf{s}_{i+1}} < n_i$.
- Let $k_{i+1} = k_i + j + 2$ and $n_{i+1} = \llbracket \mathbf{E}'[\rho] \rrbracket_{\mathbf{s}_{i+1}}$.
- Then, we have
 $\checkmark D_{k_{i+1}} = \text{while } E \text{ do } C \text{ od}$,
 $\checkmark (\mathbf{s}_{i+1}, \mathbf{h}_{i+1} \models_{\text{dom}(\mathbf{T})} \mathbf{P}[\rho] \wedge \text{word}(E) \wedge 0 < \mathbf{E}'[\rho] = n_{i+1}) \wedge \mathbf{s}_{i+1} \sim_{\mathbf{T}} s_{k_{i+1}} \wedge \mathbf{h}_{i+1} \uplus \mathbf{h}_{\mathbf{F}} \approx_{\mathbf{T}} h_{k_{i+1}}$,
 $\checkmark 0 < n_{i+1} < n_i$.

□

5.3 Soundness of Outer-level Rules

Definition 2 (Generalized triple).

$$\begin{aligned}
\{\{P\}\} C \{\{Q\}\} : k \text{ iff } & \forall j \leq k. \forall \rho \in \text{Env}(\text{FLV}(P, Q)), \mathbf{s}, \mathbf{h}, \mathbf{h}_{\mathbf{F}}, \mathbf{T}, s, h, C', s', h'. \\
& \mathbf{s}, \mathbf{h} \models P[\rho] \wedge \mathbf{s} \approx_{\mathbf{T}} s \wedge \mathbf{h} \uplus \mathbf{h}_{\mathbf{F}} \approx_{\mathbf{T}} h \wedge C, s, h \rightsquigarrow^j C', s', h' \implies \\
& (C', s', h' \rightsquigarrow -) \vee \\
& (\exists s', \mathbf{h}', \mathbf{T}'. C' = \text{skip} \wedge s', \mathbf{h}' \models Q[\rho] \wedge \\
& (\forall \mathbf{x} \notin \text{Mod}(C). s'(\mathbf{x}) = \mathbf{s}(\mathbf{x})) \wedge s' \approx_{\mathbf{T}'} s' \wedge \mathbf{h}' \uplus \mathbf{h}_{\mathbf{F}} \approx_{\mathbf{T}'} h')
\end{aligned}$$

5.3.1 Alloc

Theorem 15 (Soundness: Alloc).

$$\frac{m \geq 0}{\llbracket \mathbf{x} = 2m + 1 \rrbracket \text{alloc } \mathbf{x} \llbracket \mathbf{x} \hookrightarrow_m 0, \dots, 0 \rrbracket}$$

Proof.

- Assume: $m, \mathbf{s}, \mathbf{h}, \mathbf{h}_{\mathbf{F}}, \mathbf{T}, s, h, C', s', h'$ such that
 $\checkmark m \geq 0 \wedge \mathbf{s}, \mathbf{h} \models \mathbf{x} = 2m + 1 \wedge \mathbf{s} \approx_{\mathbf{T}} s \wedge \mathbf{h} \uplus \mathbf{h}_{\mathbf{F}} \approx_{\mathbf{T}} h \wedge \text{alloc } \mathbf{x}, s, h \rightsquigarrow^* C', s', h'$
- $\text{alloc } \mathbf{x}, s, h$ does not diverge as it takes at most one step.
- To show:
(*) $C', s', h' \rightsquigarrow -$; or
(**) $\exists s', \mathbf{h}', \mathbf{T}'. C' = \text{skip} \wedge s', \mathbf{h}' \models \mathbf{x} \hookrightarrow_m 0, \dots, 0 \wedge$
 $(\forall \mathbf{y} \notin \text{Mod}(C). s'(\mathbf{y}) = \mathbf{s}(\mathbf{y})) \wedge s' \approx_{\mathbf{T}'} s' \wedge \mathbf{h}' \uplus \mathbf{h}_{\mathbf{F}} \approx_{\mathbf{T}'} h'$

- From $\mathbf{s}, \mathbf{h} \models \mathbf{x} = 2m + 1$, we have
 $\checkmark \mathbf{s}(\mathbf{x}) = 2m + 1$.
- From $\mathbf{s}(\mathbf{x}) = 2m + 1 \wedge \mathbf{s} \approx_{\mathbf{T}} s$, we have
 $\checkmark s(\mathbf{x}) = 2m + 1$.
- From $\mathbf{h} \uplus \mathbf{h}_F \approx_{\mathbf{T}} h$, we have σ_0 such that
 $\checkmark \sigma_0 = I_{gc}(\text{dom}(\text{shape}(\mathbf{T})), h) \wedge \text{shape}(\mathbf{T}) \subseteq \sigma_0$.
- From $\mathbf{s} \approx_{\mathbf{T}} s$, we have
 $\checkmark \text{roots}(s) \subseteq \text{dom}(\text{shape}(\mathbf{T})) \subseteq \text{reach}(\text{dom}(\text{shape}(\mathbf{T})), h, \sigma_0)$.
- Thus, by GCAXiom₂, we have σ'_0 such that
 $\checkmark \sigma'_0 = I_{gc}(\text{roots}(s), h) \wedge \sigma'_0 \subseteq \sigma_0$.
- By GCAXiom₁, we have
 $\text{reach}(\text{roots}(s), h, \sigma'_0) \subseteq \text{dom}(\sigma'_0)$.
- By Lemmas 8 and 10 we have
 $\checkmark \text{reach}(\text{roots}(s), h, \sigma'_0) \subseteq \text{reach}(\text{dom}(\text{shape}(\mathbf{T})), h, \sigma_0) \subseteq \text{dom}(\text{shape}(\mathbf{T}))$
- By the specification of garbage collector, from $\text{alloc } \mathbf{x}, s, h \rightsquigarrow^* C', s', h'$ we have the following two cases.
- When $C' = \text{alloc } \mathbf{x} \wedge s' = s \wedge h' = h$:
(*) holds by the specification of garbage collector.
- When $C' = \text{skip} \wedge$
 $\checkmark \sigma_1 \uplus [p_1 \mapsto m] = I_{gc}(\text{roots}(s'), h') \wedge$
 $\checkmark s'(\mathbf{x}) = p_1 \wedge$
 $\checkmark h' = h_1 \uplus [p_1 \mapsto_m 0, \dots, 0] \wedge$
 $\checkmark (s, h, \sigma'_0) \cong ((s' \mid \mathbf{x} \mapsto 2m + 1), h_1, \sigma_1)$
for some p_1, h_1, σ_1 :
(**) is shown as follows.
- Let $s_1 = (s' \mid \mathbf{x} \mapsto 2m + 1)$.
- From $(s, h, \sigma'_0) \cong (s_1, h_1, \sigma_1)$, we have r such that
 $\checkmark r \in \text{Bij}(\text{reach}(\text{roots}(s), h, \sigma'_0), \text{reach}(\text{roots}(s_1), h_1, \sigma_1))$
 $\checkmark \forall \mathbf{y}. (s(\mathbf{y}), s_1(\mathbf{y})) \in \bar{r}$
 $\checkmark \forall (p, p') \in r. \exists n. \sigma'_0(p) = \sigma_1(p') = n \wedge \forall i < n. (h(p + 4i), h_1(p' + 4i)) \in \bar{r}$
where $\bar{r} \stackrel{\text{def}}{=} r \cup \{ (a, a) \mid a \in \text{NonPtrs} \}$.
- We define \mathbf{T}_1 as follows:
 $\checkmark \mathbf{T}_1(\ell) \stackrel{\text{def}}{=} \begin{cases} (p, n) & \text{if } \mathbf{T}(\ell) = (p', n) \wedge (p', p) \in r \\ \text{undef} & \text{otherwise} \end{cases}$
 \mathbf{T}_1 is well-defined because r is bijective.

- By definition, we have
 - ✓ $\text{dom}(\text{shape}(\mathbf{T}_1)) \subseteq \text{reach}(\text{roots}(s_1), h_1, \sigma_1)$.
- ✓ $\text{shape}(\mathbf{T}_1) \subseteq \sigma_1$ is shown as follows.
 - To have $\text{shape}(\mathbf{T}_1) \neq \text{undef}$, we need to show that $p \neq p'$ for any $(p, n) = \mathbf{T}_1(\ell)$ and $(p', n') = \mathbf{T}_1(\ell')$ with $\ell \neq \ell'$.
 - By definition of \mathbf{T}_1 , we have p'', p''' such that
 - ✓ $(p'', n) = \mathbf{T}(\ell) \wedge (p'', p) \in r \wedge (p''', n') = \mathbf{T}(\ell') \wedge (p''', p') \in r$.
 - From $\text{shape}(\mathbf{T}) \neq \text{undef} \wedge \ell \neq \ell'$, we have
 - ✓ $p'' \neq p'''$.
 - Since r is bijective, we conclude $p \neq p'$ from $(p'', p) \in r \wedge (p''', p') \in r \wedge p'' \neq p'''$.
 - Now it remains to show $\sigma_1(p) = n$ for any p, n such that
 - ✓ $\text{shape}(\mathbf{T}_1)(p) = n$.
 - By definition of $\text{shape}(\mathbf{T}_1)$ and \mathbf{T}_1 , we have ℓ, p' such that
 - ✓ $(p', n) = \mathbf{T}(\ell) \wedge (p', p) \in r$.
 - We thus have the equality

$$\begin{aligned} \sigma_1(p) &= \sigma'_0(p') && \text{(by } (p', p) \in r \text{)} \\ &= \sigma_0(p') && \text{(by } \sigma'_0 \subseteq \sigma_0 \wedge p' \in \text{reach}(\text{roots}(s), h, \sigma'_0) \subseteq \text{dom}(\sigma'_0) \text{)} \\ &= \text{shape}(\mathbf{T})(p') && \text{(by } \text{shape}(\mathbf{T}) \subseteq \sigma_0 \wedge p' \in \text{dom}(\text{shape}(\mathbf{T})) \text{)} \\ &= n. \end{aligned}$$
- ✓ $\mathbf{s} \approx_{\mathbf{T}_1} s_1$ is shown as follows.
 - $s_1(\mathbf{x}) = \text{phyv}_{\mathbf{T}_1}(\mathbf{s}(\mathbf{x})) \wedge \mathbf{s}(\mathbf{x}) \in \text{Safe}(\text{dom}(\mathbf{T}_1))$ holds since $\mathbf{s}(\mathbf{x}) = s_1(\mathbf{x}) = 2m + 1$.
 - Now we need to show that $s_1(\mathbf{y}) = \text{phyv}_{\mathbf{T}_1}(\mathbf{s}(\mathbf{y})) \wedge \mathbf{s}(\mathbf{y}) \in \text{Safe}(\text{dom}(\mathbf{T}_1))$ for any $\mathbf{y} \neq \mathbf{x}$.
 - From $\mathbf{s} \approx_{\mathbf{T}} s$, we have $\mathbf{s}(\mathbf{y}) \in \text{Safe}(\text{dom}(\mathbf{T}))$ and thus have the following two cases.
 - When $\mathbf{s}(\mathbf{y}) = a \in \text{NonPtrs}$:
 - We have $s(\mathbf{y}) = a$ from $\mathbf{s} \approx_{\mathbf{T}} s$.
 - Thus we have $s_1(\mathbf{y}) = a$ from $(s(\mathbf{y}), s_1(\mathbf{y})) \in \bar{r}$.
 - Thus we have $s_1(\mathbf{y}) = a = \text{phyv}_{\mathbf{T}_1}(\mathbf{s}(\mathbf{y})) \wedge \mathbf{s}(\mathbf{y}) = a \in \text{Safe}(\text{dom}(\mathbf{T}_1))$.
 - When $\mathbf{s}(\mathbf{y}) = \ell \hat{+} 0$ for $\ell \in \text{dom}(\mathbf{T})$:
 - We have $s(\mathbf{y}) = p$ for $(p, n) = \mathbf{T}(\ell)$ from $\mathbf{s} \approx_{\mathbf{T}} s$.
 - Thus we have $s_1(\mathbf{y}) = p'$ for p' with $(p, p') \in r$ from $(s(\mathbf{y}), s_1(\mathbf{y})) \in \bar{r}$.
 - Thus we have $\mathbf{T}_1(\ell) = (p', n)$.
 - Thus we have $s_1(\mathbf{y}) = p' = \text{phyv}_{\mathbf{T}_1}(\mathbf{s}(\mathbf{y})) \wedge \mathbf{s}(\mathbf{y}) = \ell \hat{+} 0 \in \text{Safe}(\text{dom}(\mathbf{T}_1))$.
- From $\mathbf{h} \uplus \mathbf{h}_F : \mathbf{T} \wedge \forall \ell \in \text{dom}(\mathbf{T}_1). \pi_2(\mathbf{T}_1(\ell)) = \pi_2(\mathbf{T}(\ell))$, we have
 - ✓ $\mathbf{h} \uplus \mathbf{h}_F : \mathbf{T}_1$.
- ✓ $\mathbf{h} \uplus \mathbf{h}_F :: \mathbf{T}_1 \wedge \text{phyh}_{\mathbf{T}_1}(\mathbf{h} \uplus \mathbf{h}_F) \subseteq h_1$ is shown as follows.
 - Since $\text{shape}(\mathbf{T}_1) \subseteq \sigma_1$ and $\overline{\text{dom}}(\sigma_1 \uplus [p_1 \mapsto m]) \neq \text{undef}$ by GCAXiom_1 , we have
 - ✓ $\overline{\text{dom}}(\text{shape}(\mathbf{T}_1)) \neq \text{undef}$.

- Thus it suffices to show that for any ℓ , $(p, n) = \mathbf{T}_1(\ell)$ and $i < n$, the following holds:
 $(\mathbf{h} \uplus \mathbf{h}_F)(\ell)(i) \in \text{Safe}(\text{dom}(\mathbf{T}_1)) \wedge h_1(p + 4i) = \text{phyv}_{\mathbf{T}_1}((\mathbf{h} \uplus \mathbf{h}_F)(\ell)(i)) \neq \text{undef}$
 - By definition of \mathbf{T}_1 we have p' such that
 $\checkmark (p', n) = \mathbf{T}(\ell)$ and $(p', p) \in r$.
 - From $\sigma'_0 \subseteq \sigma_0 \wedge p' \in \text{reach}(\text{roots}(s), h, \sigma'_0) \subseteq \text{dom}(\sigma'_0) \wedge \text{shape}(\mathbf{T}) \subseteq \sigma_0$, we have
 $\checkmark \sigma'_0(p') = \sigma_0(p') = \text{shape}(\mathbf{T})(p') = n$.
 - From $(p', p) \in r \wedge \sigma'_0(p') = n \wedge i < n$, we have
 $\checkmark (h(p' + 4i), h_1(p + 4i)) \in \bar{r}$.
 - $(\mathbf{h} \uplus \mathbf{h}_F)(\ell)(i) \in \text{Safe}(\text{dom}(\mathbf{T}))$ follows from $\mathbf{h} \uplus \mathbf{h}_F :: \mathbf{T}$, and thus we have two cases.
 - When $(\mathbf{h} \uplus \mathbf{h}_F)(\ell)(i) = a \in \text{NonPtrs}$:
 $\checkmark (\mathbf{h} \uplus \mathbf{h}_F)(\ell)(i) = a \in \text{Safe}(\text{dom}(\mathbf{T}_1))$.
 From $\mathbf{h} \uplus \mathbf{h}_F \approx_{\mathbf{T}} h$, we have
 $\checkmark h(p' + 4i) = \text{phyv}_{\mathbf{T}}((\mathbf{h} \uplus \mathbf{h}_F)(\ell)(i)) = a$.
 From $(h(p' + 4i), h_1(p + 4i)) \in \bar{r}$, we have
 $\checkmark h_1(p + 4i) = a = \text{phyv}_{\mathbf{T}_1}((\mathbf{h} \uplus \mathbf{h}_F)(\ell)(i)) \neq \text{undef}$
 - When $(\mathbf{h} \uplus \mathbf{h}_F)(\ell)(i) = \ell' \hat{+} 0$ for $\ell' \in \text{dom}(\mathbf{T})$:
 From $\mathbf{h} \uplus \mathbf{h}_F \approx_{\mathbf{T}} h$, we have
 $\checkmark h(p' + 4i) = \text{phyv}_{\mathbf{T}}((\mathbf{h} \uplus \mathbf{h}_F)(\ell)(i)) = p''$ for $(p'', n') = \mathbf{T}(\ell')$.
 From $(h(p' + 4i), h_1(p + 4i)) \in \bar{r}$, we have
 $\checkmark h_1(p + 4i) = p'''$ for $(p'', p''') \in r$.
 Since $\mathbf{T}_1(\ell') = (p''', n')$, we have
 $\checkmark (\mathbf{h} \uplus \mathbf{h}_F)(\ell)(i) \in \text{Safe}(\text{dom}(\mathbf{T}_1))$
 $\checkmark h_1(p + 4i) = p''' = \text{phyv}_{\mathbf{T}_1}((\mathbf{h} \uplus \mathbf{h}_F)(\ell)(i)) \neq \text{undef}$.
- Now we do case analysis on m and show (**).
 - When $m = 0$:
 - We have
 $\checkmark p_1 = 0 \wedge h' = h_1$.
 - Let
 $\checkmark \mathbf{s}' = (\mathbf{s} \mid \mathbf{x} \mapsto 0)$,
 $\checkmark \mathbf{h}' = \mathbf{h}$,
 $\checkmark \mathbf{T}' = \mathbf{T}_1$.
 - $\mathbf{s}', \mathbf{h}' \models \mathbf{x} \hookrightarrow_m 0, \dots, 0$ follows from $(\mathbf{x} \hookrightarrow_0 \epsilon) = \text{true}$.
 - $\mathbf{s}' \approx_{\mathbf{T}'}$ \mathbf{s}' follows from
 - (1) $\mathbf{s} \approx_{\mathbf{T}_1} s_1$; and
 - (2) $s'(x) = p_1 = 0 = \text{phyv}_{\mathbf{T}'}(s'(x)) \wedge s'(x) = 0 \in \text{Safe}(\text{dom}(\mathbf{T}'))$.
 - To show $\mathbf{h}' \uplus \mathbf{h}_F \approx_{\mathbf{T}'} h'$, it suffices to show $\text{shape}(\mathbf{T}') \subseteq I_{\text{gc}}(\text{dom}(\text{shape}(\mathbf{T}')), h')$ since we already have $\mathbf{h} \uplus \mathbf{h}_F : \mathbf{T}_1 \wedge \mathbf{h} \uplus \mathbf{h}_F :: \mathbf{T}_1 \wedge \text{phyh}_{\mathbf{T}_1}(\mathbf{h} \uplus \mathbf{h}_F) \subseteq h_1$.
 By GCAxiom₂, from $\sigma_1 = I_{\text{gc}}(\text{roots}(s'), h')$ and $\text{dom}(\text{shape}(\mathbf{T}')) = \text{dom}(\text{shape}(\mathbf{T}_1)) \subseteq \text{reach}(\text{roots}(s_1), h_1, \sigma_1) = \text{reach}(\text{roots}(s'), h', \sigma_1)$, we have σ_2 such that

$\checkmark \sigma_2 = I_{\text{gc}}(\text{dom}(\text{shape}(\mathbf{T}')), h') \wedge \sigma_2 \subseteq \sigma_1$.

Now it suffices to show $\text{shape}(\mathbf{T}') \subseteq \sigma_2$, which follows from

- (1) $\text{shape}(\mathbf{T}') = \text{shape}(\mathbf{T}_1) \subseteq \sigma_1 \wedge \sigma_2 \subseteq \sigma_1$; and
- (2) $\text{dom}(\text{shape}(\mathbf{T}')) \subseteq \text{reach}(\text{dom}(\text{shape}(\mathbf{T}')), h', \sigma_2) \subseteq \text{dom}(\sigma_2)$ by GCAxiom₁.

• When $m > 0$:

– Choose a fresh ℓ_1 such that $\ell_1 \notin \text{dom}(\mathbf{T}_1) \wedge \text{dom}((\mathbf{h} \uplus \mathbf{h}_F)(\ell_1)) = \emptyset$.

– Let

- $\checkmark s' = (\mathbf{s} \mid \mathbf{x} \mapsto \ell_1 \hat{+} 0)$,
- $\checkmark \mathbf{h}' = \mathbf{h} \uplus [\ell_1 \mapsto_m 0, \dots, 0]$,
- $\checkmark \mathbf{T}' = \mathbf{T}_1 \uplus [\ell_1 \mapsto (p_1, m)]$.

– $s', \mathbf{h}' \models \mathbf{x} \hookrightarrow_m 0, \dots, 0$ follows from $s'(\mathbf{x}) = \ell_1 \hat{+} 0$ and $[\ell_1 \mapsto_m 0, \dots, 0] \subseteq \mathbf{h}'$.

– $s' \approx_{\mathbf{T}'} s'$ follows from

- (1) $\mathbf{s} \approx_{\mathbf{T}_1} s_1 \wedge \mathbf{T}_1 \subseteq \mathbf{T}'$; and
- (2) $s'(\mathbf{x}) = p_1 = \text{phyv}_{\mathbf{T}'}(s'(\mathbf{x})) \wedge s'(\mathbf{x}) = \ell_1 \hat{+} 0 \in \text{Safe}(\text{dom}(\mathbf{T}'))$.

– $\mathbf{h}' \uplus \mathbf{h}_F \approx_{\mathbf{T}'} h'$ holds because

- (1) $\mathbf{h}' \uplus \mathbf{h}_F : \mathbf{T}'$ follows from $\mathbf{h} \uplus \mathbf{h}_F : \mathbf{T}_1 \wedge \text{dom}((\mathbf{h}' \uplus \mathbf{h}_F)(\ell_1)) = \{0, \dots, m-1\}$;
- (2) $\mathbf{h}' \uplus \mathbf{h}_F :: \mathbf{T}'$ follows from $\mathbf{h} \uplus \mathbf{h}_F :: \mathbf{T}_1 \wedge \forall i < m. (\mathbf{h}' \uplus \mathbf{h}_F)(\ell_1)(i) = 0 \in \text{Safe}(\text{dom}(\mathbf{T}'))$;
- (3) $\text{phyh}_{\mathbf{T}'}(\mathbf{h}' \uplus \mathbf{h}_F) \subseteq h'$ follows from $\text{phyh}_{[\ell_1 \mapsto (p_1, m)]}([\ell_1 \mapsto_m 0, \dots, 0]) = [p_1 \mapsto_m 0, \dots, 0]$ and $\text{phyh}_{\mathbf{T}_1}(\mathbf{h} \uplus \mathbf{h}_F) \subseteq h_1$; and
- (4) $\text{shape}(\mathbf{T}') \subseteq I_{\text{gc}}(\text{dom}(\text{shape}(\mathbf{T}')), h')$ is shown as follows.

Since $\text{dom}(\text{shape}(\mathbf{T}_1)) \subseteq \text{reach}(\text{roots}(s_1), h_1, \sigma_1) \subseteq \text{reach}(\text{roots}(s'), h', \sigma_1 \uplus [p_1 \mapsto m])$ holds by Lemma 8, and since $p_1 \in \text{roots}(s')$ holds, we have

$\checkmark \text{dom}(\text{shape}(\mathbf{T}')) = (\text{dom}(\text{shape}(\mathbf{T}_1)) \cup \{p_1\}) \subseteq \text{reach}(\text{roots}(s'), h', \sigma_1 \uplus [p_1 \mapsto m])$.

Thus from $\sigma_1 \uplus [p_1 \mapsto m] = I_{\text{gc}}(\text{roots}(s'), h')$, by GCAxiom₂ we have σ_2 such that

$\checkmark \sigma_2 = I_{\text{gc}}(\text{dom}(\text{shape}(\mathbf{T}')), h') \wedge \sigma_2 \subseteq \sigma_1 \uplus [p_1 \mapsto m]$.

Now it suffices to show $\text{shape}(\mathbf{T}') \subseteq \sigma_2$, which follows from

- (1) $\text{shape}(\mathbf{T}') \subseteq \sigma_1 \uplus [p_1 \mapsto m]$ by $\text{shape}(\mathbf{T}_1) \subseteq \sigma_1$;
- (2) $\sigma_2 \subseteq \sigma_1 \uplus [p_1 \mapsto m]$; and
- (3) $\text{dom}(\text{shape}(\mathbf{T}')) \subseteq \text{reach}(\text{dom}(\text{shape}(\mathbf{T}')), h', \sigma_2) \subseteq \text{dom}(\sigma_2)$ by GCAxiom₁.

□

5.3.2 Incl

Theorem 16 (Soundness: Incl).

$$\frac{V \subseteq_{\text{fin}} \text{ProgVars} \quad \{P \wedge \text{safe}(V)\} C \{Q \wedge \text{safe}(\text{Mod}(C))\}}{\{\{P\}\} C \{\{Q\}\}}$$

$$\frac{V \subseteq_{\text{fin}} \text{ProgVars} \quad [P \wedge \text{safe}(V)] C [Q \wedge \text{safe}(\text{Mod}(C))]}{[[P]] C [[Q]]}$$

Proof.

- Assume: $\forall \rho \in \text{Env}(\text{FLV}(P, Q)), \mathbf{s}, \mathbf{h}, \mathbf{h}_F, \mathbf{T}, s, h, C', s', h'$.
 $\mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} (P[\rho] \wedge \text{safe}(V)) \wedge \mathbf{s} \sim_{\mathbf{T}} s \wedge \mathbf{h} \uplus \mathbf{h}_F \approx_{\mathbf{T}} h \wedge C, s, h \rightsquigarrow^* C', s', h' \implies$
 $((C', s', h' \rightsquigarrow -) \vee$
 $(\exists \mathbf{s}', \mathbf{h}'. C' = \text{skip} \wedge \mathbf{s}', \mathbf{h}' \models_{\text{dom}(\mathbf{T})} (Q[\rho] \wedge \text{safe}(\text{Mod}(C))) \wedge$
 $(\forall y \notin \text{Mod}(C). \mathbf{s}'(y) = \mathbf{s}(y)) \wedge \mathbf{s}' \sim_{\mathbf{T}} s' \wedge \mathbf{h}' \uplus \mathbf{h}_F \approx_{\mathbf{T}} h'))$
 $[\# \wedge \neg(C, s, h \text{ diverges}) \#]$
- Assume: $\rho \in \text{Env}(\text{FLV}(P, Q)), \mathbf{s}, \mathbf{h}, \mathbf{h}_F, \mathbf{T}, s, h, C', s', h'$ such that
 $\checkmark \mathbf{s}, \mathbf{h} \models P[\rho] \wedge \mathbf{s} \approx_{\mathbf{T}} s \wedge \mathbf{h} \uplus \mathbf{h}_F \approx_{\mathbf{T}} h \wedge C, s, h \rightsquigarrow^* C', s', h'$
- To show:
 $[\# \neg(C, s, h \text{ diverges})];$ and $\#]$
 $(*) C', s', h' \rightsquigarrow -;$ or
 $(**) \exists \mathbf{s}', \mathbf{h}', \mathbf{T}'. C' = \text{skip} \wedge \mathbf{s}', \mathbf{h}' \models Q[\rho] \wedge$
 $(\forall y \notin \text{Mod}(C). \mathbf{s}'(y) = \mathbf{s}(y)) \wedge \mathbf{s}' \approx_{\mathbf{T}'} s' \wedge \mathbf{h}' \uplus \mathbf{h}_F \approx_{\mathbf{T}'} h'$
- From $\mathbf{s}, \mathbf{h} \models P[\rho]$ and $\mathbf{s} \approx_{\mathbf{T}} s$, by Lemma 6 we have
 $\checkmark \mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} P[\rho] \wedge \text{safe}(V).$
- $[\# \neg(C, s, h \text{ diverges})$ by assumption $\#]$
- Also by assumption we have two cases.
- When $C', s', h' \rightsquigarrow -$:
 $(*)$ holds.
- When $C' = \text{skip} \wedge \mathbf{s}', \mathbf{h}' \models_{\text{dom}(\mathbf{T})} (Q[\rho] \wedge \text{safe}(\text{Mod}(C))) \wedge (\forall y \notin \text{Mod}(C). \mathbf{s}'(y) = \mathbf{s}(y)) \wedge \mathbf{s}' \sim_{\mathbf{T}}$
 $s' \wedge \mathbf{h}' \uplus \mathbf{h}_F \approx_{\mathbf{T}} h'$ for some \mathbf{s}', \mathbf{h}' :
 $(**)$ is shown as follows.
- To show $(**)$, it suffices to show that $\mathbf{s}', \mathbf{h}' \models Q[\rho] \wedge \mathbf{s}' \approx_{\mathbf{T}} s'$.
- $\mathbf{s}', \mathbf{h}' \models Q[\rho]$ follows from $\mathbf{s}', \mathbf{h}' \models_{\text{dom}(\mathbf{T})} Q[\rho]$ by Lemmas 6.
- $\mathbf{s}' \approx_{\mathbf{T}} s'$ holds as follows.
 - when $\mathbf{x} \in \text{Mod}(C)$:
 $\text{phyv}_{\mathbf{T}}(\mathbf{s}'(\mathbf{x})) = s'(\mathbf{x})$ follows from $\mathbf{s}' \sim_{\mathbf{T}} s'$.
 $\mathbf{s}'(\mathbf{x}) \in \text{Safe}(\text{dom}(\mathbf{T}))$ follows from $\mathbf{s}', \mathbf{h}' \models_{\text{dom}(\mathbf{T})} \text{safe}(\text{Mod}(C)).$
 - when $\mathbf{x} \notin \text{Mod}(C)$:
 $\text{phyv}_{\mathbf{T}}(\mathbf{s}'(\mathbf{x})) = \text{phyv}_{\mathbf{T}}(\mathbf{s}(\mathbf{x})) = s(\mathbf{x}) = s'(\mathbf{x})$ follows from $\mathbf{s} \approx_{\mathbf{T}} s$ and $s(\mathbf{x}) = s'(\mathbf{x}).$
 $\mathbf{s}'(\mathbf{x}) = s(\mathbf{x}) \in \text{Safe}(\text{dom}(\mathbf{T}))$ follows from $\mathbf{s} \approx_{\mathbf{T}} s.$

□

5.3.3 Seq

Lemma 12 (Soundness: Generalized Seq).

$$\frac{\{\{P\}\} C_1 \{\{Q\}\} : k \quad \{\{Q\}\} C_2 \{\{R\}\} : k}{\{\{P\}\} C_1; C_2 \{\{R\}\} : k}$$

Proof.

- Assume: $\{\{P\}\} C_1 \{\{Q\}\} : k$
- Assume: $\{\{Q\}\} C_2 \{\{R\}\} : k$
- Assume: $\rho \in \text{Env}(\text{FLV}(P, R))$, $j, \mathbf{s}, \mathbf{h}, \mathbf{h}_F, \mathbf{T}, s, h, C', s', h'$ such that $j \leq k \wedge \mathbf{s}, \mathbf{h} \models P[\rho] \wedge \mathbf{s} \approx_{\mathbf{T}} s \wedge \mathbf{h} \uplus \mathbf{h}_F \approx_{\mathbf{T}} h \wedge (C_1; C_2, s, h \rightsquigarrow^j C', s', h')$
- To show:
 - (*) $(C', s', h' \rightsquigarrow -) \vee$
 - (**) $(\exists \mathbf{s}', \mathbf{h}', \mathbf{T}'. C' = \text{skip} \wedge \mathbf{s}', \mathbf{h}' \models R[\rho] \wedge (\forall \mathbf{y} \notin \text{Mod}(C_1; C_2). \mathbf{s}'(\mathbf{y}) = \mathbf{s}(\mathbf{y})) \wedge \mathbf{s}' \approx_{\mathbf{T}'} s' \wedge \mathbf{h}' \uplus \mathbf{h}_F \approx_{\mathbf{T}'} h')$
- Let $\rho' := \rho|^{\text{FLV}(Q)}$.
- Then, as $P[\rho] = P[\rho']$, we have $\checkmark \mathbf{s}, \mathbf{h} \models P[\rho']$.
- From $C_1; C_2, s, h \rightsquigarrow^j C', s', h'$, we have two cases.
- When $C_1, s, h \rightsquigarrow^j C'_1, s', h' \wedge C' = C'_1; C_2$:
 - By assumption we have two cases.
 - When $C'_1, s', h' \rightsquigarrow -$:
 - (*) holds because $(C'_1; C_2), s', h' \rightsquigarrow -$.
 - When $C'_1 = \text{skip} \wedge (\mathbf{s}', \mathbf{h}' \models Q[\rho']) \wedge (\forall \mathbf{y} \notin \text{Mod}(C_1). \mathbf{s}'(\mathbf{y}) = \mathbf{s}(\mathbf{y})) \wedge \mathbf{s}' \approx_{\mathbf{T}'} s' \wedge \mathbf{h}' \uplus \mathbf{h}_F \approx_{\mathbf{T}'} h'$ for some \mathbf{s}', \mathbf{h}' :
 - (*) holds because $(\text{skip}; C_2), s', h' \rightsquigarrow C_2, s', h'$.
- When $C_1, s, h \rightsquigarrow^{j_1} \text{skip}, s'_1, h'_1 \wedge C_2, s'_1, h'_1 \rightsquigarrow^{j_2} C', s', h' \wedge j = j_1 + j_2 + 1$:
 - As $j_1 \leq k \wedge C_1, s, h \rightsquigarrow^{j_1} \text{skip}, s'_1, h'_1$, by assumption we have $\mathbf{s}'_1, \mathbf{h}'_1, \mathbf{T}'_1$ such that $\checkmark \mathbf{s}'_1, \mathbf{h}'_1 \models Q[\rho'] \wedge (\forall \mathbf{y} \notin \text{Mod}(C_1). \mathbf{s}'_1(\mathbf{y}) = \mathbf{s}(\mathbf{y})) \wedge \mathbf{s}'_1 \approx_{\mathbf{T}'_1} s'_1 \wedge \mathbf{h}'_1 \uplus \mathbf{h}_F \approx_{\mathbf{T}'_1} h'_1$.
 - As $j_2 \leq k \wedge C_2, s'_1, h'_1 \rightsquigarrow^{j_2} C', s', h'$, by assumption we have two cases.
 - When $C', s', h' \rightsquigarrow -$:
 - (*) holds.
 - When $C' = \text{skip} \wedge \mathbf{s}', \mathbf{h}' \models R[\rho'] \wedge (\forall \mathbf{y} \notin \text{Mod}(C_2). \mathbf{s}'(\mathbf{y}) = \mathbf{s}'_1(\mathbf{y})) \wedge \mathbf{s}' \approx_{\mathbf{T}'} s' \wedge \mathbf{h}' \uplus \mathbf{h}_F \approx_{\mathbf{T}'} h'$:
 - (**) holds because
 - (1) $\mathbf{s}', \mathbf{h}' \models R[\rho]$ holds since $R[\rho'] = R[\rho]$;
 - (2) $(\forall \mathbf{y} \notin \text{Mod}(C_1; C_2). \mathbf{s}'(\mathbf{y}) = \mathbf{s}(\mathbf{y}))$ follows from $(\forall \mathbf{y} \notin \text{Mod}(C_2). \mathbf{s}'(\mathbf{y}) = \mathbf{s}'_1(\mathbf{y}))$ and $(\forall \mathbf{y} \notin \text{Mod}(C_1). \mathbf{s}'_1(\mathbf{y}) = \mathbf{s}(\mathbf{y}))$ since $\text{Mod}(C_1; C_2) = \text{Mod}(C_1) \cup \text{Mod}(C_2)$.

□

Theorem 17 (Soundness: Seq (partial)).

$$\frac{\{\{P\}\} C_1 \{\{Q\}\} \quad \{\{Q\}\} C_2 \{\{R\}\}}{\{\{P\}\} C_1; C_2 \{\{R\}\}}$$

Proof. It holds by Lemma 12. □

Theorem 18 (Soundness: Seq (total)).

$$\frac{[[P]] C_1 [[Q]] \quad [[Q]] C_2 [[R]]}{[[P]] C_1; C_2 [[R]]}$$

Proof.

- Assume $[[P]] C_1 [[Q]]$.
- Assume $[[Q]] C_2 [[R]]$.
- By Theorem 17, we have $\{\{P\}\} C_1; C_2 \{\{R\}\}$.
- Assume: $\rho \in \text{Env}(\text{FLV}(P, R))$, $\mathbf{s}, \mathbf{h}, \mathbf{h}_F, \mathbf{T}, s, h$ such that $\mathbf{s}, \mathbf{h} \models P[\rho] \wedge \mathbf{s} \approx_{\mathbf{T}} s \wedge \mathbf{h} \uplus \mathbf{h}_F \approx_{\mathbf{T}} h$.
- Now we show $\neg(C_1; C_2, s, h \text{ diverges})$ by contradiction.
- Assume $\{D_i, s_i, h_i\}_{i \in \mathbb{N}}$ such that $\checkmark (D_0, s_0, h_0) = (C_1; C_2, s, h) \wedge \forall i. D_i, s_i, h_i \rightsquigarrow D_{i+1}, s_{i+1}, h_{i+1}$.
- Let $\rho' := \rho|^{\text{FLV}(Q)}$.
- Then, as $P[\rho] = P[\rho']$, we have $\mathbf{s}, \mathbf{h} \models P[\rho']$.
- By $[[P]] C_1 [[Q]]$, we have $\neg(C_1, s, h \text{ diverges})$.
- Thus, we have some k such that $D_k = (\text{skip}; C_2)$ and $C_1, s, h \rightsquigarrow^k \text{skip}, s_k, h_k$.
- As $D_k = (\text{skip}; C_2)$, we have $D_{k+1} = C_2$, $s_{k+1} = s_k$, and $h_{k+1} = h_k$.
- By $[[P]] C_1 [[Q]]$, we have $\mathbf{s}', \mathbf{h}', \mathbf{T}'$ such that $\checkmark \mathbf{s}', \mathbf{h}' \models Q[\rho'] \wedge (\forall y \notin \text{Mod}(C_1). \mathbf{s}'(y) = \mathbf{s}(y)) \wedge \mathbf{s}' \approx_{\mathbf{T}'} s_k \wedge \mathbf{h}' \uplus \mathbf{h}_F \approx_{\mathbf{T}'} h_k$.
- By $[[Q]] C_2 [[R]]$, we have $\neg(C_2, s_k, h_k \text{ diverges})$.
- Thus we have $\neg(D_{k+1}, s_{k+1}, h_{k+1} \text{ diverges})$, which is a contradiction.

□

5.3.4 Frame

Theorem 19 (Soundness: Frame).

$$\frac{\{\{P\}\} C \{\{Q\}\} \quad \text{FPV}(R) \cap \text{Mod}(C) = \emptyset}{\{\{P * R\}\} C \{\{Q * R\}\}} \quad \frac{[[P]] C [[Q]] \quad \text{FPV}(R) \cap \text{Mod}(C) = \emptyset}{[[P * R]] C [[Q * R]]}$$

Proof.

- Assume: $\text{FPV}(R) \cap \text{Mod}(C) = \emptyset$
- Assume: $\forall \rho \in \text{Env}(\text{FLV}(P, Q)), \mathbf{s}, \mathbf{h}, \mathbf{h}_F, \mathbf{T}, s, h, C', s', h'$.
 $\mathbf{s}, \mathbf{h} \models P[\rho] \wedge \mathbf{s} \approx_{\mathbf{T}} s \wedge \mathbf{h} \uplus \mathbf{h}_F \approx_{\mathbf{T}} h \wedge C, s, h \rightsquigarrow^* C', s', h' \implies$
 $((C', s', h' \rightsquigarrow -) \vee$
 $(\exists \mathbf{s}', \mathbf{h}', \mathbf{T}'. C' = \text{skip} \wedge \mathbf{s}', \mathbf{h}' \models Q[\rho] \wedge$
 $(\forall \mathbf{y} \notin \text{Mod}(C). \mathbf{s}'(\mathbf{y}) = \mathbf{s}(\mathbf{y})) \wedge \mathbf{s}' \approx_{\mathbf{T}'} s' \wedge \mathbf{h}' \uplus \mathbf{h}_F \approx_{\mathbf{T}'} h'))$
 $[\# \wedge \neg(C, s, h \text{ diverges}) \#]$
- Assume: $\rho \in \text{Env}(\text{FLV}(P, Q, R)), \mathbf{s}, \mathbf{h}, \mathbf{h}_F, \mathbf{T}, s, h, C', s', h'$ such that
 $\mathbf{s}, \mathbf{h} \models (P[\rho] * R[\rho]) \wedge \mathbf{s} \approx_{\mathbf{T}} s \wedge \mathbf{h} \uplus \mathbf{h}_F \approx_{\mathbf{T}} h \wedge C, s, h \rightsquigarrow^* C', s', h'$
- To show:
 $[\# \neg(C, s, h \text{ diverges})];$ and $\#]$
 $(*) (C', s', h' \rightsquigarrow -) \vee$
 $(**) (\exists \mathbf{s}', \mathbf{h}', \mathbf{T}'. C' = \text{skip} \wedge \mathbf{s}', \mathbf{h}' \models (Q[\rho] * R[\rho]) \wedge$
 $(\forall \mathbf{y} \notin \text{Mod}(C). \mathbf{s}'(\mathbf{y}) = \mathbf{s}(\mathbf{y})) \wedge \mathbf{s}' \approx_{\mathbf{T}'} s' \wedge \mathbf{h}' \uplus \mathbf{h}_F \approx_{\mathbf{T}'} h')$
- From $\mathbf{s}, \mathbf{h} \models (P[\rho] * R[\rho])$, we have \mathbf{h}_1 and \mathbf{h}_2 such that
 $\checkmark \mathbf{h} = \mathbf{h}_1 \uplus \mathbf{h}_2,$
 $\checkmark \mathbf{s}, \mathbf{h}_1 \models P[\rho],$
 $\checkmark \mathbf{s}, \mathbf{h}_2 \models R[\rho].$
- $[\# \neg(C, s, h \text{ diverges})$ holds by assumption since $\mathbf{h} \uplus \mathbf{h}_F = \mathbf{h}_1 \uplus (\mathbf{h}_2 \uplus \mathbf{h}_F) \wedge \mathbf{s}, \mathbf{h}_1 \models P[\rho] \#]$
- Also by assumption we have two cases since $\mathbf{h} \uplus \mathbf{h}_F = \mathbf{h}_1 \uplus (\mathbf{h}_2 \uplus \mathbf{h}_F) \wedge \mathbf{s}, \mathbf{h}_1 \models P[\rho].$
- When $C', s', h' \rightsquigarrow -$:
 $(*)$ holds.
- When $C' = \text{skip} \wedge (\mathbf{s}', \mathbf{h}' \models Q[\rho]) \wedge (\forall \mathbf{y} \notin \text{Mod}(C). \mathbf{s}'(\mathbf{y}) = \mathbf{s}(\mathbf{y})) \wedge \mathbf{s}' \approx_{\mathbf{T}'} s' \wedge \mathbf{h}' \uplus \mathbf{h}_2 \uplus \mathbf{h}_F \approx_{\mathbf{T}'} h'$
for some \mathbf{s}', \mathbf{h}' :
 $(**)$ is shown as follows.
- To show $(**)$, it suffices to show that $\mathbf{s}', \mathbf{h}' \uplus \mathbf{h}_2 \models Q[\rho] * R[\rho].$
- We split the heap $\mathbf{h}' \uplus \mathbf{h}_2$ into \mathbf{h}' and \mathbf{h}_2 .
- As $\mathbf{s}', \mathbf{h}' \models Q[\rho]$ holds, we need to show $\mathbf{s}', \mathbf{h}_2 \models R[\rho]$, which follows from $(\mathbf{s}, \mathbf{h}_2 \models R[\rho]) \wedge (\forall \mathbf{y} \notin \text{Mod}(C). \mathbf{s}'(\mathbf{y}) = \mathbf{s}(\mathbf{y})) \wedge \text{FPV}(R) \cap \text{Mod}(C) = \emptyset$ by Lemma 5.

□

5.3.5 Conseq

Theorem 20 (Soundness: Conseq).

$$\frac{P \models P' \quad \{\{P'\}\} C \{\{Q'\}\} \quad Q' \models Q}{\{\{P\}\} C \{\{Q\}\}} \quad \frac{P \models P' \quad [[P']] C [[Q']] \quad Q' \models Q}{[[P]] C [[Q]}}$$

Proof.

- Assume: $P \models P'$ and $Q' \models Q$.
- Assume: $\forall \rho \in \text{Env}(\text{FLV}(P', Q')), \mathbf{s}, \mathbf{h}, \mathbf{h}_F, \mathbf{T}, s, h, C', s', h'$.
 $\mathbf{s}, \mathbf{h} \models P'[\rho] \wedge \mathbf{s} \approx_{\mathbf{T}} s \wedge \mathbf{h} \uplus \mathbf{h}_F \approx_{\mathbf{T}} h \wedge C, s, h \rightsquigarrow^* C', s', h' \implies$
 $((C', s', h' \rightsquigarrow -) \vee$
 $(\exists \mathbf{s}', \mathbf{h}', \mathbf{T}'. C' = \text{skip} \wedge \mathbf{s}', \mathbf{h}' \models Q'[\rho] \wedge$
 $(\forall \mathbf{y} \notin \text{Mod}(C). \mathbf{s}'(\mathbf{y}) = \mathbf{s}(\mathbf{y})) \wedge \mathbf{s}' \approx_{\mathbf{T}'} s' \wedge \mathbf{h}' \uplus \mathbf{h}_F \approx_{\mathbf{T}'} h'))$
 $[\# \wedge \neg(C, s, h \text{ diverges}) \#]$
- Assume: $\rho \in \text{Env}(\text{FLV}(P, Q)), \mathbf{s}, \mathbf{h}, \mathbf{h}_F, \mathbf{T}, s, h, C', s', h'$ such that
 $\mathbf{s}, \mathbf{h} \models P[\rho] \wedge \mathbf{s} \approx_{\mathbf{T}} s \wedge \mathbf{h} \uplus \mathbf{h}_F \approx_{\mathbf{T}} h \wedge C, s, h \rightsquigarrow^* C', s', h'$
- To show:
 $[\# \wedge \neg(C, s, h \text{ diverges})];$ and $\#]$
 $(*) (C', s', h' \rightsquigarrow -) \vee$
 $(**) (\exists \mathbf{s}', \mathbf{h}', \mathbf{T}'. C' = \text{skip} \wedge \mathbf{s}', \mathbf{h}' \models Q[\rho] \wedge$
 $(\forall \mathbf{y} \notin \text{Mod}(C). \mathbf{s}'(\mathbf{y}) = \mathbf{s}(\mathbf{y})) \wedge \mathbf{s}' \approx_{\mathbf{T}'} s' \wedge \mathbf{h}' \uplus \mathbf{h}_F \approx_{\mathbf{T}'} h')$
- Let $\rho' \stackrel{\text{def}}{=} \rho|_{\text{FLV}(P', Q')}$.
- From $P \models P'$ and $\mathbf{s} \approx_{\mathbf{T}} s \wedge \mathbf{h} \uplus \mathbf{h}_F \approx_{\mathbf{T}} h \wedge \mathbf{s}, \mathbf{h} \models P[\rho']$ (as $P[\rho'] = P[\rho]$), we have
 $\checkmark \mathbf{s}, \mathbf{h} \models P'[\rho']$.
- $[\# \wedge \neg(C, s, h \text{ diverges})]$ holds by assumption. $\#]$
- Also by assumption we have two cases.
- When $C', s', h' \rightsquigarrow -$:
 $(*)$ holds.
- When $C' = \text{skip} \wedge (s', h' Q'[\rho']) \wedge (\forall \mathbf{y} \notin \text{Mod}(C). \mathbf{s}'(\mathbf{y}) = \mathbf{s}(\mathbf{y})) \wedge \mathbf{s}' \approx_{\mathbf{T}'} s' \wedge \mathbf{h}' \uplus \mathbf{h}_F \approx_{\mathbf{T}'} h'$ for some \mathbf{s}', \mathbf{h}' :
 $(**)$ holds because $\mathbf{s}', \mathbf{h}' \models Q[\rho]$ follows from $Q' \models Q$ and $\mathbf{s}' \approx_{\mathbf{T}'} s' \wedge \mathbf{h}' \uplus \mathbf{h}_F \approx_{\mathbf{T}'} h' \wedge \mathbf{s}', \mathbf{h}' \models Q'[\rho']$ (as $Q[\rho'] = Q[\rho]$).

□

5.3.6 Ex

Theorem 21 (Soundness: Ex).

$$\frac{\{\{P\}\} C \{\{Q\}\}}{\{\{\exists v. P\}\} C \{\{\exists v. Q\}\}} \quad \frac{[[P]] C [[Q]]}{[[\exists v. P]] C [[\exists v. Q]]}$$

Proof.

- Assume: $\forall \rho \in \text{Env}(\text{FLV}(P, Q)), \mathbf{s}, \mathbf{h}, \mathbf{h}_F, \mathbf{T}, s, h, C', s', h'$.
 $\mathbf{s}, \mathbf{h} \models P[\rho] \wedge \mathbf{s} \approx_{\mathbf{T}} s \wedge \mathbf{h} \uplus \mathbf{h}_F \approx_{\mathbf{T}} h \wedge C, s, h \rightsquigarrow^* C', s', h' \implies$
 $((C', s', h' \rightsquigarrow -) \vee$
 $(\exists \mathbf{s}', \mathbf{h}', \mathbf{T}'. C' = \text{skip} \wedge \mathbf{s}', \mathbf{h}' \models Q[\rho] \wedge$
 $(\forall \mathbf{y} \notin \text{Mod}(C). \mathbf{s}'(\mathbf{y}) = \mathbf{s}(\mathbf{y})) \wedge \mathbf{s}' \approx_{\mathbf{T}'} s' \wedge \mathbf{h}' \uplus \mathbf{h}_F \approx_{\mathbf{T}'} h'))$
 $[\# \wedge \neg(C, s, h \text{ diverges}) \#]$
- Assume: $\rho \in \text{Env}(\text{FLV}(\exists v. P, \exists v. Q)), \mathbf{s}, \mathbf{h}, \mathbf{h}_F, \mathbf{T}, s, h, C', s', h'$ such that
 $(\mathbf{s}, \mathbf{h} \models (\exists v. P)[\rho]) \wedge \mathbf{s} \approx_{\mathbf{T}} s \wedge \mathbf{h} \uplus \mathbf{h}_F \approx_{\mathbf{T}} h \wedge C, s, h \rightsquigarrow^* C', s', h'$
- To show:
 $[\# \neg(C, s, h \text{ diverges})]$ and $\#]$
 $(*) (C', s', h' \rightsquigarrow -) \vee$
 $(**) (\exists \mathbf{s}', \mathbf{h}', \mathbf{T}'. C' = \text{skip} \wedge \mathbf{s}', \mathbf{h}' \models (\exists v. Q)[\rho] \wedge$
 $(\forall \mathbf{y} \notin \text{Mod}(C). \mathbf{s}'(\mathbf{y}) = \mathbf{s}(\mathbf{y})) \wedge \mathbf{s}' \approx_{\mathbf{T}'} s' \wedge \mathbf{h}' \uplus \mathbf{h}_F \approx_{\mathbf{T}'} h')$
- From $\mathbf{s}, \mathbf{h} \models (\exists v. P)[\rho]$, by Lemma 7 we have
 $\checkmark \mathbf{s}, \mathbf{h} \models P[(\rho \mid v \mapsto \mathbf{v})]$ for some $\mathbf{v} \in \text{LogVals}$.
- Let $\rho' := (\rho \mid v \mapsto \mathbf{v})$.
- $[\# \neg(C, s, h \text{ diverges})]$ holds by assumption. $\#]$
- Also by assumption we have two cases.
- When $C', s', h' \rightsquigarrow -$:
 $(*)$ holds.
- When $C' = \text{skip} \wedge (s', h' \models Q[\rho']) \wedge (\forall \mathbf{y} \notin \text{Mod}(C). \mathbf{s}'(\mathbf{y}) = \mathbf{s}(\mathbf{y})) \wedge \mathbf{s}' \approx_{\mathbf{T}'} s' \wedge \mathbf{h}' \uplus \mathbf{h}_F \approx_{\mathbf{T}'} h'$
for some \mathbf{s}', \mathbf{h}' :
 $(**)$ holds because $\mathbf{s}', \mathbf{h}' \models (\exists v. Q)[\rho]$ follows from $\mathbf{s}', \mathbf{h}' \models Q[\rho']$ by Lemma 7.

□

5.3.7 Gen

Theorem 22 (Soundness: Gen).

$$\frac{\forall \mathbf{v} \in \text{LogVals}. \{\{P[\mathbf{v}/v]\}\} C \{\{Q[\mathbf{v}/v]\}\}}{\{\{P\}\} C \{\{Q\}\}} \quad \frac{\forall \mathbf{v} \in \text{LogVals}. [[P[\mathbf{v}/v]]] C [[Q[\mathbf{v}/v]]]}{[[P]] C [[Q]]}$$

Proof. The goal directly follows by definition because $P[\rho] = P[\rho(v)/v][\rho]$ and $Q[\rho] = Q[\rho(v)/v][\rho]$ for any $\rho \in \text{Env}(\text{FLV}(P, Q))$. □

5.3.8 Total

Theorem 23 (Soundness: Total).

$$\frac{[[P]] C [[Q]]}{\{\{P\}\} C \{\{Q\}\}}$$

Proof. It holds vacuously by definition. □

5.3.9 If

Theorem 24 (Soundness: If).

$$\frac{\{\{P \wedge E\}\} C_1 \{\{Q\}\} \quad \{\{P \wedge \text{not } E\}\} C_2 \{\{Q\}\}}{\{\{P \wedge \text{word}(E)\}\} \text{ if } E \text{ then } C_1 \text{ else } C_2 \text{ fi } \{\{Q\}\}} \quad \frac{[[P \wedge E]] C_1 [[Q]] \quad [[P \wedge \text{not } E]] C_2 [[Q]]}{[[P \wedge \text{word}(E)]] \text{ if } E \text{ then } C_1 \text{ else } C_2 \text{ fi } [[Q]]}$$

Proof.

- Assume: $\forall \rho \in \text{Env}(\text{FLV}(P, Q)), \mathbf{s}, \mathbf{h}, \mathbf{h}_F, \mathbf{T}, s, h, C', s', h'$.
 $(\mathbf{s}, \mathbf{h} \models P[\rho] \wedge E) \wedge \mathbf{s} \approx_{\mathbf{T}} s \wedge \mathbf{h} \uplus \mathbf{h}_F \approx_{\mathbf{T}} h \wedge C_1, s, h \rightsquigarrow^* C', s', h' \implies$
 $((C', s', h' \rightsquigarrow -) \vee$
 $(\exists \mathbf{s}', \mathbf{h}', \mathbf{T}'. C' = \text{skip} \wedge \mathbf{s}', \mathbf{h}' \models Q[\rho] \wedge$
 $(\forall \mathbf{y} \notin \text{Mod}(C_1). \mathbf{s}'(\mathbf{y}) = \mathbf{s}(\mathbf{y})) \wedge \mathbf{s}' \approx_{\mathbf{T}'} s' \wedge \mathbf{h}' \uplus \mathbf{h}_F \approx_{\mathbf{T}'} h'))$
 $[\# \wedge \neg(C_1, s, h \text{ diverges}) \#]$
- Assume: $\forall \rho \in \text{Env}(\text{FLV}(P, Q)), \mathbf{s}, \mathbf{h}, \mathbf{h}_F, \mathbf{T}, s, h, C', s', h'$.
 $(\mathbf{s}, \mathbf{h} \models P[\rho] \wedge \text{not } E) \wedge \mathbf{s} \approx_{\mathbf{T}} s \wedge \mathbf{h} \uplus \mathbf{h}_F \approx_{\mathbf{T}} h \wedge C_2, s, h \rightsquigarrow^* C', s', h' \implies$
 $((C', s', h' \rightsquigarrow -) \vee$
 $(\exists \mathbf{s}', \mathbf{h}', \mathbf{T}'. C' = \text{skip} \wedge \mathbf{s}', \mathbf{h}' \models Q[\rho] \wedge$
 $(\forall \mathbf{y} \notin \text{Mod}(C_2). \mathbf{s}'(\mathbf{y}) = \mathbf{s}(\mathbf{y})) \wedge \mathbf{s}' \approx_{\mathbf{T}'} s' \wedge \mathbf{h}' \uplus \mathbf{h}_F \approx_{\mathbf{T}'} h'))$
 $[\# \wedge \neg(C_2, s, h \text{ diverges}) \#]$
- Assume: $\rho \in \text{Env}(\text{FLV}(P, Q)), \mathbf{s}, \mathbf{h}, \mathbf{h}_F, \mathbf{T}, s, h, C', s', h'$ such that
 $(\mathbf{s}, \mathbf{h} \models P[\rho] \wedge \text{word}(E)) \wedge \mathbf{s} \approx_{\mathbf{T}} s \wedge \mathbf{h} \uplus \mathbf{h}_F \approx_{\mathbf{T}} h \wedge \text{if } E \text{ then } C_1 \text{ else } C_2 \text{ fi}, s, h \rightsquigarrow^* C', s', h'$
- To show:
 $[\# \neg(\text{if } E \text{ then } C_1 \text{ else } C_2 \text{ fi}, s, h \text{ diverges}); \text{ and } \#]$
 $(*) (C', s', h' \rightsquigarrow -) \vee$
 $(**) (\exists \mathbf{s}', \mathbf{h}', \mathbf{T}'. C' = \text{skip} \wedge \mathbf{s}', \mathbf{h}' \models Q[\rho] \wedge$
 $(\forall \mathbf{y} \notin \text{Mod}(C_1, C_2). \mathbf{s}'(\mathbf{y}) = \mathbf{s}(\mathbf{y})) \wedge \mathbf{s}' \approx_{\mathbf{T}'} s' \wedge \mathbf{h}' \uplus \mathbf{h}_F \approx_{\mathbf{T}'} h')$
- From $\mathbf{s}, \mathbf{h} \models \text{word}(E)$, we have
 $\checkmark \llbracket E \rrbracket_{\mathbf{s}} \in \text{Words}$.
- By Lemma 2, we have
 $\checkmark \llbracket E \rrbracket_s = \text{phyv}_{\mathbf{T}}(\llbracket E \rrbracket_{\mathbf{s}}) = \llbracket E \rrbracket_{\mathbf{s}}$.
- Thus, we have two cases.
- When $\llbracket E \rrbracket_{\mathbf{s}} \in \text{Words} \setminus \{0\}$:
 - $[\#$ Since we have $\text{if } E \text{ then } C_1 \text{ else } C_2 \text{ fi}, s, h \rightsquigarrow C_1, s, h$ and $\mathbf{s}, \mathbf{h} \models P[\rho] \wedge E$, by assumption we have $\neg(C_1, s, h \text{ diverges})$ and thus $\neg(C, s, h \text{ diverges})$ holds. $\#]$
 - From $\text{if } E \text{ then } C_1 \text{ else } C_2 \text{ fi}, s, h \rightsquigarrow^* C', s', h'$ we have two cases.
 - When $C' = \text{if } E \text{ then } C_1 \text{ else } C_2 \text{ fi} \wedge s' = s \wedge h' = h$:
 $(*)$ holds as we have $\text{if } E \text{ then } C_1 \text{ else } C_2 \text{ fi}, s, h \rightsquigarrow C_1, s, h$.
 - When $C_1, s, h \rightsquigarrow^* C', s', h'$:
 $(*)$ or $(**)$ holds by assumption since we have $\mathbf{s}, \mathbf{h} \models P[\rho] \wedge E$.

- When $\llbracket E \rrbracket_{\mathbf{s}} = 0$:
 - $\#[\#]$ Since we have if E then C_1 else C_2 fi, $s, h \rightsquigarrow C_2, s, h$ and $\mathbf{s}, \mathbf{h} \models P[\rho] \wedge \text{not } E$, by assumption we have $\neg(C_2, s, h \text{ diverges})$ and thus $\neg(C, s, h \text{ diverges})$ holds. $\#$
 - From if E then C_1 else C_2 fi, $s, h \rightsquigarrow^* C', s', h'$ we have two cases.
 - When $C' = \text{if } E \text{ then } C_1 \text{ else } C_2 \text{ fi} \wedge s' = s \wedge h' = h$:
 - (*) holds as we have if E then C_1 else C_2 fi, $s, h \rightsquigarrow C_2, s, h$.
 - When $C_2, s, h \rightsquigarrow^* C', s', h'$:
 - (*) or (**) holds by assumption since we have $\mathbf{s}, \mathbf{h} \models P[\rho] \wedge \text{not } E$.

□

5.3.10 While

Theorem 25 (Soundness: While).

$$\frac{\{\{P \wedge E\}\} C \{\{P \wedge \text{word}(E)\}\}}{\{\{P \wedge \text{word}(E)\}\} \text{ while } E \text{ do } C \text{ od } \{\{P \wedge \text{not } E\}\}}$$

Proof.

- Assume: $\{\{P \wedge E\}\} C \{\{P \wedge \text{word}(E)\}\}$
- To show: $\forall k. \{\{P \wedge \text{word}(E)\}\} \text{ while } E \text{ do } C \text{ od } \{\{P \wedge \text{not } E\}\} : k$
- We prove the goal by induction on k .
- (Base case) when $k = 0$,
 - Assume: $\rho \in \text{Env}(\text{FLV}(P)), \mathbf{s}, \mathbf{h}, \mathbf{h}_F, \mathbf{T}, s, h, C', s', h'$ such that $(\mathbf{s}, \mathbf{h} \models P[\rho] \wedge \text{word}(E)) \wedge \mathbf{s} \approx_{\mathbf{T}} s \wedge \mathbf{h} \uplus \mathbf{h}_F \approx_{\mathbf{T}} h \wedge \text{while } E \text{ do } C \text{ od}, s, h \rightsquigarrow^k C', s', h'$.
 - It suffices to show
 - (*) $C', s', h' \rightsquigarrow -$.
 - From $\mathbf{s}, \mathbf{h} \models \text{word}(E)$, we have $\checkmark \llbracket E \rrbracket_{\mathbf{s}} \in \text{Words}$.
 - By Lemma 2, we have $\checkmark \llbracket E \rrbracket_s = \text{phyv}_{\mathbf{T}}(\llbracket E \rrbracket_{\mathbf{s}}) = \llbracket E \rrbracket_{\mathbf{s}}$.
 - (*) holds because $C' = \text{while } E \text{ do } C \text{ od} \wedge s' = s \wedge h' = h$ and $\llbracket E \rrbracket_s \neq \text{undef}$.
- (Inductive step) when $k > 0 \wedge \forall j < k. \{\{P \wedge \text{word}(E)\}\} \text{ while } E \text{ do } C \text{ od } \{\{P \wedge \text{not } E\}\} : j$,
 - Assume: $\rho \in \text{Env}(\text{FLV}(P)), \mathbf{s}, \mathbf{h}, \mathbf{h}_F, \mathbf{T}, s, h, C', s', h'$ such that $(\mathbf{s}, \mathbf{h} \models P[\rho] \wedge \text{word}(E)) \wedge \mathbf{s} \approx_{\mathbf{T}} s \wedge \mathbf{h} \uplus \mathbf{h}_F \approx_{\mathbf{T}} h \wedge \text{while } E \text{ do } C \text{ od}, s, h \rightsquigarrow^k C', s', h'$.
 - To show:
 - (*) $(C', s', h' \rightsquigarrow -) \vee$
 - (**) $(\exists \mathbf{s}', \mathbf{h}', \mathbf{T}'. C' = \text{skip} \wedge (\mathbf{s}', \mathbf{h}' \models P[\rho] \wedge \text{not } E) \wedge (\forall \mathbf{y} \notin \text{Mod}(C). \mathbf{s}'(\mathbf{y}) = \mathbf{s}(\mathbf{y})) \wedge \mathbf{s}' \approx_{\mathbf{T}'} s' \wedge \mathbf{h}' \uplus \mathbf{h}_F \approx_{\mathbf{T}'} h')$

- From $\mathbf{s}, \mathbf{h} \models \text{word}(E)$, we have
 - ✓ $\llbracket E \rrbracket_{\mathbf{s}} \in \text{Words}$.
- By Lemma 2, we have
 - ✓ $\llbracket E \rrbracket_s = \text{phyv}_{\mathbf{T}}(\llbracket E \rrbracket_{\mathbf{s}}) = \llbracket E \rrbracket_{\mathbf{s}}$.
- Thus we have two cases.
- When $\llbracket E \rrbracket_s = \llbracket E \rrbracket_{\mathbf{s}} = 0$:
 - ◇ We have $\text{while } E \text{ do } C \text{ od}, s, h \rightsquigarrow \text{skip}, s, h$.
 - ◇ Thus we have $\text{skip}, s, h \rightsquigarrow^{k-1} C', s', h'$, from which it follows that
 - ✓ $C' = \text{skip} \wedge s' = s \wedge h' = h$.
 - ◇ Thus $(**)$ holds because we have $\mathbf{s}, \mathbf{h} \models \text{not } E$ from $\llbracket \text{not } E \rrbracket_{\mathbf{s}} = 1$.
- When $\llbracket E \rrbracket_s = \llbracket E \rrbracket_{\mathbf{s}} \in \text{Words} \setminus \{0\}$:
 - ◇ We have $\text{while } E \text{ do } C \text{ od}, s, h \rightsquigarrow (C; \text{while } E \text{ do } C \text{ od}), s, h$, from which we have
 - ✓ $(C; \text{while } E \text{ do } C \text{ od}), s, h \rightsquigarrow^{k-1} C', s', h'$.
 - ◇ From $\{\{P \wedge E\}\} C \{\{P \wedge \text{word}(E)\}\}$ and $\{\{P \wedge \text{word}(E)\}\} \text{while } E \text{ do } C \text{ od} \{\{P \wedge \text{not } E\}\} : k - 1$, by Lemma 12 we have
 - ✓ $\{\{P \wedge E\}\} C; \text{while } E \text{ do } C \text{ od} \{\{P \wedge \text{not } E\}\} : k - 1$.
 - ◇ Thus $(*) \vee (**)$ holds since we have $\mathbf{s}, \mathbf{h} \models E$ from $\llbracket E \rrbracket_{\mathbf{s}} \in \text{Words} \setminus \{0\}$.

□

Theorem 26 (Soundness: WhileT).

$$\frac{\llbracket [P \wedge E \wedge 0 < \mathbf{E}' = v] C \llbracket [P \wedge \text{word}(E) \wedge 0 < \mathbf{E}' < v] \rrbracket \quad v \notin \text{FLV}(P, \mathbf{E}')}{\llbracket [P \wedge \text{word}(E) \wedge 0 < \mathbf{E}'] \rrbracket \text{while } E \text{ do } C \text{ od} \llbracket [P \wedge \text{not } E] \rrbracket}}$$

Proof.

- Assume: $\llbracket [P \wedge E \wedge 0 < \mathbf{E}' = v] C \llbracket [P \wedge \text{word}(E) \wedge 0 < \mathbf{E}' < v] \rrbracket$ and $v \notin \text{FLV}(P, \mathbf{E}')$.
- By Theorems 20, 21, 23 and 25, we have

$$\begin{array}{c} \frac{\llbracket [P \wedge E \wedge 0 < \mathbf{E}' = v] C \llbracket [P \wedge \text{word}(E) \wedge 0 < \mathbf{E}' < v] \rrbracket}{\llbracket [\exists v. P \wedge E \wedge 0 < \mathbf{E}' = v] C \llbracket [\exists v. P \wedge \text{word}(E) \wedge 0 < \mathbf{E}' < v] \rrbracket \rrbracket} \text{ (Ex)} \\ \frac{\llbracket [P \wedge 0 < \mathbf{E}' \wedge E] C \llbracket [P \wedge 0 < \mathbf{E}' \wedge \text{word}(E)] \rrbracket}{\llbracket \{\{P \wedge 0 < \mathbf{E}' \wedge E\}\} C \{\{P \wedge 0 < \mathbf{E}' \wedge \text{word}(E)\}\} \rrbracket} \text{ (Total)} \\ \frac{\llbracket \{\{P \wedge 0 < \mathbf{E}' \wedge \text{word}(E)\}\} \text{while } E \text{ do } C \text{ od} \{\{P \wedge 0 < \mathbf{E}' \wedge \text{not } E\}\} \rrbracket}{\llbracket [P \wedge \text{word}(E) \wedge 0 < \mathbf{E}'] \rrbracket \text{while } E \text{ do } C \text{ od} \llbracket [P \wedge \text{not } E] \rrbracket \rrbracket} \text{ (While)} \\ \text{ (Conseq)} \end{array}$$

- Assume: $\rho \in \text{Env}(\text{FLV}(P, \mathbf{E}')), \mathbf{s}, \mathbf{h}, \mathbf{h}_{\mathbf{F}}, \mathbf{T}, s, h$ such that
 - $(\mathbf{s}, \mathbf{h} \models P[\rho] \wedge \text{word}(E) \wedge 0 < \mathbf{E}'[\rho]) \wedge \mathbf{s} \approx_{\mathbf{T}} s \wedge \mathbf{h} \uplus \mathbf{h}_{\mathbf{F}} \approx_{\mathbf{T}} h$.
- Now we show $\neg(\text{while } E \text{ do } C \text{ od}, s, h \text{ diverges})$ by contradiction.
- Assume: $\{D_i, s_i, h_i\}_{i \in \mathbb{N}}$ such that
 - ✓ $(D_0, s_0, h_0) = (\text{while } E \text{ do } C \text{ od}, s, h) \wedge \forall i. D_i, s_i, h_i \rightsquigarrow D_{i+1}, s_{i+1}, h_{i+1}$.

- We show the following, which is a contradiction because $n_0 > n_1 > n_2 \dots > 0$ is not possible.
- By induction on i , we find $\{k_i, n_i, \mathbf{s}_i, \mathbf{h}_i, \mathbf{T}_i\}_{i \in \mathbb{N}}$ (with $n_i \in \text{Words}$) such that
 - ✓ $D_{k_i} = \text{while } E \text{ do } C \text{ od}$;
 - ✓ $(\mathbf{s}_i, \mathbf{h}_i \models P[\rho] \wedge \text{word}(E) \wedge 0 < \mathbf{E}'[\rho] = n_i) \wedge \mathbf{s}_i \approx_{\mathbf{T}_i} s_{k_i} \wedge \mathbf{h}_i \uplus \mathbf{h}_F \approx_{\mathbf{T}_i} h_{k_i}$;
 - ✓ if $i > 0$ then $0 < n_i < n_{i-1}$.

(Base Case)

- From $(\mathbf{s}, \mathbf{h} \models 0 < \mathbf{E}'[\rho])$, we have
 - ✓ $\llbracket \mathbf{E}'[\rho] \rrbracket_{\mathbf{s}} \in \text{Words}$.
- Let $k_0 = 0$, $\mathbf{s}_0 = \mathbf{s}$, $\mathbf{h}_0 = \mathbf{h}$, $\mathbf{T}_0 = \mathbf{T}$ and $n_0 = \llbracket \mathbf{E}'[\rho] \rrbracket_{\mathbf{s}_0} \in \text{Words}$.
- Then by assumption we have
 - ✓ $D_{k_0} = \text{while } E \text{ do } C \text{ od}$,
 - ✓ $(\mathbf{s}_0, \mathbf{h}_0 \models P[\rho] \wedge \text{word}(E) \wedge 0 < \mathbf{E}'[\rho] = n_0) \wedge \mathbf{s}_0 \approx_{\mathbf{T}_0} s_{k_0} \wedge \mathbf{h}_0 \uplus \mathbf{h}_F \approx_{\mathbf{T}_0} h_{k_0}$.

(Inductive step)

- Assume:
 - ✓ $D_{k_i} = \text{while } E \text{ do } C \text{ od}$,
 - ✓ $(\mathbf{s}_i, \mathbf{h}_i \models P[\rho] \wedge \text{word}(E) \wedge 0 < \mathbf{E}'[\rho] = n_i) \wedge \mathbf{s}_i \approx_{\mathbf{T}_i} s_{k_i} \wedge \mathbf{h}_i \uplus \mathbf{h}_F \approx_{\mathbf{T}_i} h_{k_i}$.
- As $(D_{k_i}, s_{k_i}, h_{k_i})$ diverges, we have
 - ✓ $\llbracket E \rrbracket_{s_{k_i}} \in \text{Words} \setminus \{0\}$,
 - ✓ $(D_{k_{i+1}}, s_{k_{i+1}}, h_{k_{i+1}}) = (C; \text{while } E \text{ do } C \text{ od}, s_{k_i}, h_{k_i})$.
- From $\mathbf{s}_i, \mathbf{h}_i \models \text{word}(E)$, we have
 - ✓ $\llbracket E \rrbracket_{\mathbf{s}_i} \in \text{Words}$.
- By Lemma 2, we have $\llbracket E \rrbracket_{\mathbf{s}_i} = \text{phyv}_{\mathbf{T}_i}(\llbracket E \rrbracket_{\mathbf{s}_i}) = \llbracket E \rrbracket_{s_{k_i}} \in \text{Words} \setminus \{0\}$, and thus we have
 - ✓ $\mathbf{s}_i, \mathbf{h}_i \models E$.
- By $[[P \wedge E \wedge 0 < \mathbf{E}' = v]] C [[P \wedge \text{word}(E) \wedge 0 < \mathbf{E}' < v]]$, we have
 - ✓ $\neg(C, s_{k_{i+1}}, h_{k_{i+1}} \text{ diverges})$.
- Thus, we have some j such that
 - ✓ $D_{k_i+j+1} = (\text{skip}; \text{while } E \text{ do } C \text{ od})$,
 - ✓ $C, s_{k_{i+1}}, h_{k_{i+1}} \rightsquigarrow^j \text{skip}, s_{k_i+j+1}, h_{k_i+j+1}$.
- Then, by $[[P \wedge E \wedge 0 < \mathbf{E}' = v]] C [[P \wedge \text{word}(E) \wedge 0 < \mathbf{E}' < v]]$, we have $\mathbf{s}_{i+1}, \mathbf{h}_{i+1}, \mathbf{T}_{i+1}$ such that
 - ✓ $(\mathbf{s}_{i+1}, \mathbf{h}_{i+1} \models P[\rho] \wedge \text{word}(E) \wedge 0 < \mathbf{E}'[\rho] < n_i) \wedge \mathbf{s}_{i+1} \approx_{\mathbf{T}_{i+1}} s_{k_i+j+1} \wedge \mathbf{h}_{i+1} \uplus \mathbf{h}_F \approx_{\mathbf{T}_{i+1}} h_{k_i+j+1}$.
- Also we have
 - ✓ $(D_{k_i+j+2}, s_{k_i+j+2}, h_{k_i+j+2}) = (\text{while } E \text{ do } C \text{ od}, s_{k_i+j+1}, h_{k_i+j+1})$.
- From $\mathbf{s}_{i+1}, \mathbf{h}_{i+1} \models 0 < \mathbf{E}'[\rho] < n_i$, we have
 - ✓ $\llbracket \mathbf{E}'[\rho] \rrbracket_{\mathbf{s}_{i+1}} \in \text{Words} \wedge 0 < \llbracket \mathbf{E}'[\rho] \rrbracket_{\mathbf{s}_{i+1}} < n_i$.
- Let $k_{i+1} = k_i + j + 2$ and $n_{i+1} = \llbracket \mathbf{E}'[\rho] \rrbracket_{\mathbf{s}_{i+1}}$.

- Then, we have
 - ✓ $D^{k_{i+1}} = \text{while } E \text{ do } C \text{ od}$,
 - ✓ $(\mathbf{s}_{i+1}, \mathbf{h}_{i+1} \models P[\rho] \wedge \text{word}(E) \wedge 0 < \mathbf{E}'[\rho] = n_{i+1}) \wedge \mathbf{s}_{i+1} \approx_{\mathbf{T}_{i+1}} s_{k_{i+1}} \wedge \mathbf{h}_{i+1} \uplus \mathbf{h}_F \approx_{\mathbf{T}_{i+1}} h_{k_{i+1}}$,
 - ✓ $0 < n_{i+1} < n_i$.

□

5.4 Soundness of Assertion Entailments

5.4.1 NPtrSafe

Theorem 27 (NPtrSafe).

$$\text{nonptr}(\mathbf{E}) \models \text{safe}(\mathbf{E})$$

Proof. It holds vacuously by definition. □

5.4.2 BoolWord

Theorem 28 (BoolWord).

$$\mathbf{E} \models \text{word}(\mathbf{E})$$

Proof.

- For any $\rho \in \text{Env}(\text{FLV}(\mathbf{E}, \mathbf{E}'))$, $\mathbf{s}, \mathbf{h}, \mathbf{h}_F, \mathbf{T}, s, h$ such that $\mathbf{s} \sim_{\mathbf{T}} s \wedge \mathbf{h} \uplus \mathbf{h}_F \approx_{\mathbf{T}} h$, we need to show that $\mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} \mathbf{E}[\rho] \implies \mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} \text{word}(\mathbf{E}[\rho])$.
- From $\mathbf{s}, \mathbf{h} \models_{\mathbf{T}} \mathbf{E}[\rho]$, we have $\llbracket \mathbf{E}[\rho] \rrbracket_{\mathbf{s}} \in \text{Words} \setminus \{0\} \subseteq \text{Words}$.
- Thus $\mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} \text{word}(\mathbf{E}[\rho])$ holds.

□

5.4.3 PointstoNZero

Theorem 29 (PointstoNZero).

$$\mathbf{E} \leftrightarrow \mathbf{E}' \models \mathbf{E} \neq 0$$

Proof.

- For any $\rho \in \text{Env}(\text{FLV}(\mathbf{E}, \mathbf{E}'))$, $\mathbf{s}, \mathbf{h}, \mathbf{h}_F, \mathbf{T}, s, h$ such that $\mathbf{s} \sim_{\mathbf{T}} s \wedge \mathbf{h} \uplus \mathbf{h}_F \approx_{\mathbf{T}} h$, we need to show that $\mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} \mathbf{E}[\rho] \leftrightarrow \mathbf{E}'[\rho] \implies \mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} \text{word}(\mathbf{E}[\rho]) = 0$.
- From $\mathbf{s}, \mathbf{h} \models_{\mathbf{T}} \mathbf{E}[\rho] \leftrightarrow \mathbf{E}'[\rho]$, we have $\llbracket \mathbf{E}[\rho] \rrbracket_{\mathbf{s}} = \ell \hat{+} 4i$ and $\llbracket \mathbf{E}'[\rho] \rrbracket_{\mathbf{s}} = \mathbf{h}(\ell)(i)$ for some $\ell \in \text{dom}(\mathbf{T})$ and $i \in \mathbb{Z}$.
- As $\mathbf{h}(\ell)(i) \neq \text{undef}$, from $\mathbf{h} \uplus \mathbf{h}_F : \mathbf{T}$ we have $0 \leq i < n$ for $(p, n) = \mathbf{T}(\ell)$.
- Thus $\llbracket \mathbf{E}[\rho] \neq 0 \rrbracket_{\mathbf{s}} = \llbracket \text{not } (\ell \hat{+} 4i = 0) \rrbracket_{\mathbf{s}} = 1$ as $i \geq 0$.
- Thus $\mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} \mathbf{E}[\rho] \neq 0$ holds.

□

5.4.4 ExpSafe

Theorem 30 (ExpSafe).

$$\text{defined}(E) \models \text{offsafe}(E)$$

Proof.

- For any $\mathbf{s}, \mathbf{h}, \mathbf{h}_F, \mathbf{T}, s, h$ such that $\mathbf{s} \sim_{\mathbf{T}} s \wedge \mathbf{h} \uplus \mathbf{h}_F \approx_{\mathbf{T}} h$, we need to show that $\mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} \text{defined}(E) \implies \mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} \text{offsafe}(E)$.
- From $\mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} \text{defined}(E)$, we have two cases.
- When $\llbracket E \rrbracket_{\mathbf{s}} = w \in \text{Words}$:
By definition $\mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} \text{offsafe}(E)$ holds.
- When $\llbracket E \rrbracket_{\mathbf{s}} = \ell \hat{+} i$ for some $\ell \in \text{Locs}$ and $i \in \mathbb{Z}$:
By Corollary 3, we have $\ell \in \text{dom}(\mathbf{T})$ and thus $\mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} \text{offsafe}(E)$ holds.

□

5.4.5 HeapSafe

Theorem 31 (HeapSafe).

$$\mathbf{E} \hookrightarrow \mathbf{E}' \wedge \text{offsafe}(\mathbf{E}) \models \text{safe}(\mathbf{E}')$$

Proof.

- For any $\rho \in \text{Env}(\text{FLV}(\mathbf{E}, \mathbf{E}')), \mathbf{s}, \mathbf{h}, \mathbf{h}_F, \mathbf{T}, s, h$ such that $\mathbf{s} \sim_{\mathbf{T}} s \wedge \mathbf{h} \uplus \mathbf{h}_F \approx_{\mathbf{T}} h$, we need to show that $\mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} \mathbf{E} \hookrightarrow \mathbf{E}' \wedge \text{offsafe}(\mathbf{E}) \implies \mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} \text{safe}(\mathbf{E}'[\rho])$.
- From $\mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} \mathbf{E} \hookrightarrow \mathbf{E}' \wedge \text{offsafe}(\mathbf{E})$, we have $\llbracket \mathbf{E}[\rho] \rrbracket_{\mathbf{s}} = \ell \hat{+} 4i$ and $\llbracket \mathbf{E}'[\rho] \rrbracket_{\mathbf{s}} = \mathbf{h}(\ell)(i)$ for some $\ell \in \text{dom}(\mathbf{T})$ and $i \in \mathbb{Z}$.
- As $\mathbf{h}(\ell)(i) \neq \text{undef}$, from $\mathbf{h} \uplus \mathbf{h}_F : \mathbf{T}$ we have $0 \leq i < n$ for $(p, n) = \mathbf{T}(\ell)$.
- From $\mathbf{h} \uplus \mathbf{h}_F :: \mathbf{T}$, we have $\llbracket \mathbf{E}'[\rho] \rrbracket_{\mathbf{s}} = \mathbf{h}(\ell)(i) \in \text{Safe}(\text{dom}(\mathbf{T}))$.
- Thus $\mathbf{s}, \mathbf{h} \models_{\text{dom}(\mathbf{T})} \text{safe}(\mathbf{E}'[\rho])$ holds.

□

5.4.6 ExpHeapSafe

Corollary 13 (ExpHeapSafe).

$$E \hookrightarrow \mathbf{E}' \models \text{safe}(\mathbf{E}')$$

Proof. It follows as a corollary from (ExpSafe) and (HeapSafe).

□

5.4.7 SafeEq

Theorem 32 (SafeEq).

$$\text{safe}(\mathbf{E}, \mathbf{E}') \models \text{defined}(\mathbf{E} = \mathbf{E}')$$

Proof. It is obvious by definition.

□

5.5 Soundness of Derived Rules

5.5.1 Ex'

Theorem 33 (Soundness: Ex').

For $(\langle \cdot \rangle, \mathcal{P}, \mathcal{Q}) \in \{ (\{\cdot\}, \mathbf{P}, \mathbf{Q}), ([\cdot], \mathbf{P}, \mathbf{Q}), (\{\{\cdot\}\}, P, Q), ([[\cdot]], P, Q) \}$,

$$\frac{\langle \mathcal{P} \rangle C \langle \mathcal{Q} \rangle \quad v \notin \text{FLV}(\mathcal{Q})}{\langle \exists v. \mathcal{P} \rangle C \langle \mathcal{Q} \rangle}$$

Proof.

$$\frac{\frac{\langle \mathcal{P} \rangle C \langle \mathcal{Q} \rangle}{\langle \exists v. \mathcal{P} \rangle C \langle \exists v. \mathcal{Q} \rangle} \text{ (Ex)} \quad v \notin \text{FLV}(\mathcal{Q})}{\langle \exists v. \mathcal{P} \rangle C \langle \mathcal{Q} \rangle} \text{ (Conseq)}$$

□

5.5.2 Disj

Theorem 34 (Soundness: Disj).

For $(\langle \cdot \rangle, \mathcal{P}_1, \mathcal{P}_2, \mathcal{Q}) \in \{ (\{\cdot\}, \mathbf{P}_1, \mathbf{P}_2, \mathbf{Q}), ([\cdot], \mathbf{P}_1, \mathbf{P}_2, \mathbf{Q}), (\{\{\cdot\}\}, P_1, P_2, Q), ([[\cdot]], P_1, P_2, Q) \}$,

$$\frac{\langle \mathcal{P}_1 \rangle C \langle \mathcal{Q} \rangle \quad \langle \mathcal{P}_2 \rangle C \langle \mathcal{Q} \rangle}{\langle \mathcal{P}_1 \vee \mathcal{P}_2 \rangle C \langle \mathcal{Q} \rangle}$$

Proof. Choose a fresh variable u such that $u \notin \text{FLV}(\mathcal{P}_1, \mathcal{P}_2, \mathcal{Q})$.

$$\frac{\frac{\frac{\langle \mathcal{P}_1 \rangle C \langle \mathcal{Q} \rangle \quad \langle \mathcal{P}_2 \rangle C \langle \mathcal{Q} \rangle}{\forall \mathbf{v} \in \text{LogVals. } \langle (\mathbf{v} = 1 \wedge \mathcal{P}_1) \vee (\mathbf{v} = 2 \wedge \mathcal{P}_2) \rangle C \langle \mathcal{Q} \rangle} \text{ (Gen)}}{\langle (u = 1 \wedge \mathcal{P}_1) \vee (u = 2 \wedge \mathcal{P}_2) \rangle C \langle \mathcal{Q} \rangle} \text{ (Ex')}}{\langle \exists u. (u = 1 \wedge \mathcal{P}_1) \vee (u = 2 \wedge \mathcal{P}_2) \rangle C \langle \mathcal{Q} \rangle} \text{ (Conseq)} \\ \langle \mathcal{P}_1 \vee \mathcal{P}_2 \rangle C \langle \mathcal{Q} \rangle$$

□

5.5.3 Inst

Lemma 14.

For $(\langle \cdot \rangle, \mathcal{P}, \mathcal{Q}) \in \{ (\{\cdot\}, \mathbf{P}, \mathbf{Q}), ([\cdot], \mathbf{P}, \mathbf{Q}), (\{\{\cdot\}\}, P, Q), ([[\cdot]], P, Q) \}$,

$$\frac{\langle \mathcal{P} \rangle C \langle \mathcal{Q} \rangle \quad \text{FPV}(\mathbf{E}) \cap \text{Mod}(C) = \emptyset \quad v \notin \text{FLV}(\mathbf{E})}{\langle \mathcal{P}[\mathbf{E}/v] \wedge \text{defined}(\mathbf{E}) \rangle C \langle \mathcal{Q}[\mathbf{E}/v] \rangle}$$

Proof. Assume: $\langle \mathcal{P} \rangle C \langle \mathcal{Q} \rangle \wedge \text{FPV}(\mathbf{E}) \cap \text{Mod}(C) = \emptyset \wedge v \notin \text{FLV}(\mathbf{E})$.

$$\begin{aligned} & \langle \mathcal{P}[\mathbf{E}/v] \wedge \text{defined}(\mathbf{E}) \rangle \\ & \langle \exists v. \mathcal{P}[\mathbf{E}/v] * v = \mathbf{E} \rangle \\ & \langle \mathcal{P}[\mathbf{E}/v] * v = \mathbf{E} \rangle \quad \text{(Ex')} \\ & \langle \mathcal{P} * v = \mathbf{E} \rangle \end{aligned}$$

$$\begin{array}{l}
C \\
\langle \mathcal{Q} * v = \mathbf{E} \rangle \\
\langle \mathcal{Q}[\mathbf{E}/v] * v = \mathbf{E} \rangle \\
\langle \mathcal{Q}[\mathbf{E}/v] \rangle
\end{array}$$

□

Theorem 35 (Soundness: Inst).

For $(\langle \cdot \rangle, \mathcal{P}, \mathcal{Q}) \in \{(\{\cdot\}, \mathbf{P}, \mathbf{Q}), ([\cdot], \mathbf{P}, \mathbf{Q}), (\{\{\cdot\}\}, P, Q), ([[\cdot]], P, Q)\}$,

$$\frac{\langle \mathcal{P} \rangle C \langle \mathcal{Q} \rangle \quad \text{FPV}(\mathbf{E}) \cap \text{Mod}(C) = \emptyset}{\langle \mathcal{P}[\mathbf{E}/v] \wedge \text{defined}(\mathbf{E}) \rangle C \langle \mathcal{Q}[\mathbf{E}/v] \rangle}$$

Proof. Choose a fresh variable u such that $u \notin \text{FLV}(\mathcal{P}, \mathcal{Q}, \mathbf{E}, v)$.

$$\begin{array}{c}
\frac{\langle \mathcal{P} \rangle C \langle \mathcal{Q} \rangle}{\langle \mathcal{P}[u/v] \wedge \text{defined}(u) \rangle C \langle \mathcal{Q}[u/v] \rangle} \text{ (Lemma 14)} \\
\frac{\langle \mathcal{P}[u/v] \wedge \text{defined}(u) \rangle C \langle \mathcal{Q}[u/v] \rangle}{\langle \mathcal{P}[u/v] \rangle C \langle \mathcal{Q}[u/v] \rangle} \text{ (Conseq)} \\
\frac{\langle \mathcal{P}[u/v] \rangle C \langle \mathcal{Q}[u/v] \rangle}{\langle \mathcal{P}[u/v][\mathbf{E}/u] \wedge \text{defined}(\mathbf{E}) \rangle C \langle \mathcal{Q}[u/v][\mathbf{E}/u] \rangle} \text{ (Lemma 14)} \\
\frac{\langle \mathcal{P}[u/v][\mathbf{E}/u] \wedge \text{defined}(\mathbf{E}) \rangle C \langle \mathcal{Q}[u/v][\mathbf{E}/u] \rangle}{\langle \mathcal{P}[\mathbf{E}/v] \wedge \text{defined}(\mathbf{E}) \rangle C \langle \mathcal{Q}[\mathbf{E}/v] \rangle} \text{ (Conseq)}
\end{array}$$

□

5.5.4 Assign'

Theorem 36 (Soundness: Assign').

$$\overline{[\mathbf{P}[E/x] \wedge \text{defined}(E)] \mathbf{x} := E [\mathbf{P}]}$$

Proof. Choose a fresh variable v such that $v \notin \text{FLV}(\mathbf{P})$.

$$\begin{array}{l}
\langle \mathbf{P}[E/x] \wedge \text{defined}(E) \rangle \\
\langle \exists v. \mathbf{P}[E/x] \wedge \text{defined}(E) \wedge \mathbf{x} = v \rangle \\
\langle \mathbf{P}[E/x] \wedge \text{defined}(E) \wedge \mathbf{x} = v \rangle \quad (\text{Ex}') \\
\langle \mathbf{P}[E[v/x]/x] * (\text{defined}(E) \wedge \mathbf{x} = v) \rangle \\
\mathbf{x} := E \\
\langle \mathbf{P}[E[v/x]/x] * (\mathbf{x} = E[v/x]) \rangle \quad (\text{Assign}) \\
\langle \mathbf{P} \wedge \mathbf{x} = E[v/x] \rangle \\
\langle \mathbf{P} \rangle
\end{array}$$

□

5.5.5 Read' and Read''

Theorem 37 (Soundness: Read'').

$$\frac{x \notin \text{FPV}(\mathbf{E}') \cup \text{FPV}(\mathbf{E}'')}{[x = \mathbf{E}' \wedge E \leftrightarrow \mathbf{E}''] \quad x := [E] \quad [x = \mathbf{E}'' \wedge E[\mathbf{E}'/x] \leftrightarrow \mathbf{E}'']}$$

Proof. Assume: $x \notin \text{FPV}(\mathbf{E}') \cup \text{FPV}(\mathbf{E}'')$.

Choose fresh variables u, v such that $u, v \notin \text{FLV}(\mathbf{E}', \mathbf{E}'') \wedge u \neq v$.

$$\frac{\frac{\frac{[x = u \wedge E \leftrightarrow v] \quad x := [E] \quad [x = v \wedge E[u/x] \leftrightarrow v]}{\text{(Read)}}}{[x = \mathbf{E}' \wedge E \leftrightarrow \mathbf{E}'' \wedge \text{defined}(\mathbf{E}') \wedge \text{defined}(\mathbf{E}'')] \quad x := [E] \quad [x = \mathbf{E}'' \wedge E[\mathbf{E}'/x] \leftrightarrow \mathbf{E}'']}{\text{(Inst)}}}{[x = \mathbf{E}' \wedge E \leftrightarrow \mathbf{E}''] \quad x := [E] \quad [x = \mathbf{E}'' \wedge E[\mathbf{E}'/x] \leftrightarrow \mathbf{E}'']}{\text{(Conseq)}}$$

□

Theorem 38 (Soundness: Read').

$$\frac{x \notin \text{FPV}(E) \cup \text{FPV}(\mathbf{E}')}{[E \leftrightarrow \mathbf{E}'] \quad x := [E] \quad [x = \mathbf{E}' \wedge E \leftrightarrow \mathbf{E}']}$$

Proof. Assume: $x \notin \text{FPV}(E) \cup \text{FPV}(\mathbf{E}')$.

Choose a fresh name v such that $v \notin \text{FLV}(\mathbf{E}')$.

$$\frac{\frac{\frac{[x = v \wedge E \leftrightarrow \mathbf{E}'] \quad x := [E] \quad [x = \mathbf{E}' \wedge E \leftrightarrow \mathbf{E}']}{\text{(Read')}}}{[\exists v. x = v \wedge E \leftrightarrow \mathbf{E}'] \quad x := [E] \quad [x = \mathbf{E}' \wedge E \leftrightarrow \mathbf{E}']}{\text{(Ex')}}}{[E \leftrightarrow \mathbf{E}'] \quad x := [E] \quad [x = \mathbf{E}' \wedge E \leftrightarrow \mathbf{E}']}{\text{(Conseq)}}$$

□

5.5.6 ASSIGN and ASSIGN'

Theorem 39 (Soundness: ASSIGN).

$$\frac{[[\mathbf{P}[y/x]]] \quad x := y \quad [[\mathbf{P}]]}{[[\mathbf{P}]]}$$

Proof.

$$\frac{\frac{[[\mathbf{P}[y/x]]] \quad [\mathbf{P}[y/x] \wedge \text{safe}(y)]}{\text{(Incl)}}}{[\mathbf{P}[y/x] \wedge \text{safe}(y) \wedge \text{defined}(y)] \quad x := y}{\text{(Assign')}}}{[\mathbf{P} \wedge \text{safe}(x)] \quad [[\mathbf{P}]]}{\text{(Incl)}}$$

□

Theorem 40 (Soundness: ASSIGN').

$$\frac{}{[[\mathbf{P}[E/x] \wedge \text{nonptr}(E)]] \ x := E \ [[\mathbf{P}]]}$$

Proof.

$$\begin{array}{l} [[\mathbf{P}[E/x] \wedge \text{nonptr}(E)]] \\ [\mathbf{P}[E/x] \wedge \text{nonptr}(E)] \quad (\text{Incl}) \\ [\mathbf{P}[E/x] \wedge \text{nonptr}(E) \wedge \text{defined}(E)] \\ \mathbf{x} := E \\ [\mathbf{P} \wedge \text{nonptr}(\mathbf{x})] \quad (\text{Assign}') \\ [\mathbf{P} \wedge \text{safe}(\mathbf{x})] \\ [[\mathbf{P}]] \quad (\text{Incl}) \end{array}$$

□

5.5.7 READ and READ'

Theorem 41 (Soundness: READ).

$$\frac{\mathbf{x} \notin \text{FPV}(E) \cup \text{FPV}(\mathbf{E}')}{[[E \leftrightarrow \mathbf{E}']] \ \mathbf{x} := [E] \ [[\mathbf{x} = \mathbf{E}' \wedge E \leftrightarrow \mathbf{E}']]}$$

Proof. Assume: $\mathbf{x} \notin \text{FPV}(E) \cup \text{FPV}(\mathbf{E}')$.

$$\begin{array}{l} [[E \leftrightarrow \mathbf{E}']] \\ [E \leftrightarrow \mathbf{E}'] \quad (\text{Incl}) \\ \mathbf{x} := [E] \\ [\mathbf{x} = \mathbf{E}' \wedge E \leftrightarrow \mathbf{E}'] \quad (\text{Read}') \\ [\mathbf{x} = \mathbf{E}' \wedge E \leftrightarrow \mathbf{E}' \wedge \text{safe}(\mathbf{x})] \\ [[\mathbf{x} = \mathbf{E}' \wedge E \leftrightarrow \mathbf{E}']] \quad (\text{Incl}) \end{array}$$

□

Theorem 42 (Soundness: READ').

$$\frac{\mathbf{x} \notin \text{FPV}(\mathbf{E}') \cup \text{FPV}(\mathbf{E}'')}{[[\mathbf{x} = \mathbf{E}' \wedge E \leftrightarrow \mathbf{E}'']] \ \mathbf{x} := [E] \ [[\mathbf{x} = \mathbf{E}'' \wedge E[\mathbf{E}'/x] \leftrightarrow \mathbf{E}'']]}$$

Proof. Assume: $\mathbf{x} \notin \text{FPV}(\mathbf{E}') \cup \text{FPV}(\mathbf{E}'')$.

$$\begin{array}{l} [[\mathbf{x} = \mathbf{E}' \wedge E \leftrightarrow \mathbf{E}'']] \\ [\mathbf{x} = \mathbf{E}' \wedge E \leftrightarrow \mathbf{E}'''] \quad (\text{Incl}) \\ [\mathbf{x} = \mathbf{E}' \wedge E \leftrightarrow \mathbf{E}'' * \text{safe}(\mathbf{E}'')] \\ \mathbf{x} := [E] \end{array}$$

$$[x = \mathbf{E}'' \wedge E[\mathbf{E}'/x] \hookrightarrow \mathbf{E}'' * \text{safe}(\mathbf{E}'')] \quad (\text{Read}'')$$

$$[x = \mathbf{E}'' \wedge E[\mathbf{E}'/x] \hookrightarrow \mathbf{E}'' \wedge \text{safe}(x)]$$

$$[[x = \mathbf{E}'' \wedge E[\mathbf{E}'/x] \hookrightarrow \mathbf{E}''']] \quad (\text{Incl})$$

□

5.5.8 WRITE and WRITE'

Theorem 43 (Soundness: WRITE).

$$\frac{}{[[E \hookrightarrow -]] [E] := x [[E \hookrightarrow x]]}$$

Proof.

$$\begin{array}{l} [[E \hookrightarrow -]] \\ [E \hookrightarrow - \wedge \text{safe}(x)] \end{array} \quad (\text{Incl})$$

$$\begin{array}{l} [E] := x \\ [E \hookrightarrow x] \end{array} \quad (\text{Write})$$

$$[[E \hookrightarrow x]] \quad (\text{Incl})$$

□

Theorem 44 (Soundness: WRITE').

$$\frac{}{[[E \hookrightarrow - \wedge \text{nonptr}(E')]] [E] := E' [[E \hookrightarrow E']]}$$

Proof.

$$\begin{array}{l} [[E \hookrightarrow - \wedge \text{nonptr}(E')]] \\ [E \hookrightarrow - \wedge \text{nonptr}(E')] \end{array} \quad (\text{Incl})$$

$$\begin{array}{l} [E \hookrightarrow - \wedge \text{safe}(E')] \\ [E] := E' \end{array}$$

$$[E \hookrightarrow E'] \quad (\text{Write})$$

$$[[E \hookrightarrow E']] \quad (\text{Incl})$$

□

5.5.9 ALLOC

Theorem 45 (Soundness: ALLOC).

$$\frac{n \geq 0}{[[E = 2n + 1]] x := \text{ALLOC}(E) [[x \hookrightarrow_n 0, \dots, 0]]}$$

Proof. Assume: $n \geq 0$.

$[[E = 2n + 1]]$

$[[E = 2n + 1 \wedge \text{nonptr}(E)]]$

$x := E;$

$[[x = 2n + 1]]$

(ASSIGN')

alloc x

$[[x \hookrightarrow_n 0, \dots, 0]]$

(Alloc)

□