2 Logical Relation

2.1 Recursive Domain Equation

The goal is to solve the following domain equation:

$$\mathrm{Wor} = \mathbb{N} \xrightarrow{fin} (\mathrm{State} \times \mathrm{Rel} \times (\mathrm{State} \to (\mathrm{Wor} \xrightarrow{\text{"mon, ne"}} \mathrm{UPred}(\mathrm{HeapSegment}))))$$

Where State is a set of states with all the ones we use in this paper.

$$\mathrm{Rel} = \{R \in \mathcal{P}(\mathrm{State}^2) \mid R \text{ is reflexive and transitive}\}$$

This cannot be solved with sets, so we use preordered complete ordered families of equivalences where it is possible to solve such an equation that ressembles the above one, namely it is possible to find an isomorphism ξ and preordered c.o.f.e. W such that

$$\xi : \mathrm{Wor} \cong \blacktriangleright(\mathbb{N} \xrightarrow{fin} (\mathrm{State} \times \mathrm{Rel} \times (\mathrm{State} \to (\mathrm{Wor} \xrightarrow{\text{"mon, ne"}} \mathrm{UPred}(\mathrm{HeapSegment})))))$$

Definition 1 (o.f.e's). *An ordered family of equivalences (o.f.e.) is a set and a family of equivalences, $\left(X, \left(\stackrel{n}{=}\right)_{n=0}^{\infty}\right)$. The familly of equivalences have to satisfy the following properties*

- $\stackrel{0}{=}$ *is a total relation on* X
- $\forall n. \forall x, y \in S. x \stackrel{n+1}{=} y \Rightarrow x \stackrel{n}{=} y$
- $\forall x, y. (\forall n. x \stackrel{n}{=} y) \Rightarrow x = y$

■

DD: I suppose you're using a standard ultrametric metric to make an o.f.e. a metric space?

Definition 2 (c.o.f.e.'s). *A complete orderede family of equivalences is an o.f.e. $\left(X, \left(\stackrel{n}{=}\right)_{n=0}^{\infty}\right)$ where all Cauchy sequences in X have a limit in X.* ■

Definition 3 (Preordered c.o.f.e.'s). *A preordered c.o.f.e. is a c.o.f.e. equiped with a preorder on X, $\left(X, \left(\stackrel{n}{=}\right)_{n=0}^{\infty}, \sqsupseteq\right)$.*

- *The ordering preserves limits. That is, for Cauchy chains $\{a_n\}_n$ and $\{b_n\}_n$ in X if $\{a_n\}_n \sqsupseteq \{b_n\}_n$, then $\lim\{a_n\}_n \sqsupseteq \lim\{b_n\}_n$.*

■

Definition 4 (Preordered c.o.f.e. construction: Finite-partial function). *Given a set S and preordered c.o.f.e. X, $S \xrightarrow{fin} X$ is a preordered c.o.f.e. with the ordering*

$$f \sqsupseteq g$$
$$\textit{iff}$$
$$\mathrm{dom}(f) \supseteq \mathrm{dom}(g) \;\; \textit{and} \;\; \forall n \in S. f(n) \sqsupseteq g(n)$$

■

We need the following constructions to create the preordered c.o.f.e. needed to solve the recursive domain equation. DD: this sentence doesn't parse :)

Definition 5 (Preordered c.o.f.e. construction: Function). *Given a set S and c.o.f.e. HP, $S \to HP$ is a preordered c.o.f.e. with the ordering*

$$f \sqsupseteq g$$
$$\textit{iff}$$
$$\forall s \in \mathrm{dom}(f).\, f(s) \sqsupseteq g(s)$$

■

Definition 6 (Preordered c.o.f.e. construction: Monotone, non-expansive function). *Given a preordered c.o.f.e. W and preordered c.o.f.e. U, $W \xrightarrow{\text{mon, ne}} U$ is a preordered c.o.f.e. with the ordering*

$$f \sqsupseteq g$$
$$\textit{iff}$$
$$\forall s \in \mathrm{dom}(f).\, f(s) \sqsupseteq g(s)$$

■

The above are standard constructions, so they are used here without showing they are in fact well-defined as shown in Birkedal and Bizjak [2014].

Definition 7 (Preordered c.o.f.e. construction: Region). *Given a c.o.f.e. H, the tuple*

$$(\mathrm{State} \times \mathrm{Rel} \times H)$$

is a preordered c.o.f.e. with the ordering

$$(s_2, \phi_2, H_2) \sqsupseteq (s_1, \phi_1, H_2)$$
$$\textit{iff}$$
$$H_2 = H_1 \textit{ and } \phi_2 = \phi_1 \textit{ and } (s_1, s_2) \in \phi_2$$

■

Lemma 2 (Region definition well-defined). *The construction in Definition 7 is a preordered c.o.f.e.. That is*

- *It is a c.o.f.e. (this is a standard construction)*
- *$\sqsupseteq$ is a transitive and reflexive relation.*
- *$\sqsupseteq$ preserves limits.*
  That is for Cauchy chains $\{a_n\}_n$ and $\{b_n\}_n$ if

  $$\{a_n\}_n \geq \{b_n\}_n,$$

  then

  $$\lim\{a_n\}_n \sqsupseteq \lim\{b_n\}_n$$

■

The category of c.o.f.e.'s is the category whith c.o.f.e.'s as objects and non-expansive functions as morphisms. We denote this category $\mathbb{C}$. The category of preordered c.o.f.e.'s has preordered c.o.f.e.'s as objects and monotone and non-expansive functions as morphisms. We denote this category $\mathbb{P}$.

Define functors K, R, and G as follows:

$$K : \mathbb{P} \to \mathbb{P}$$
$$K(R) = \mathbb{N} \stackrel{fin}{\rightharpoonup} R$$
$$K(f) = \lambda\phi.\, \lambda n.\, f(\phi(n))$$

$$R : \mathbb{C} \to \mathbb{P}$$
$$R(H) = \mathrm{State} \times \mathrm{Rel} \times H$$
$$R(h) = \lambda(s, \Phi, H).\, (s, \Phi, h(H))$$

$$G : \mathbb{P}^{op} \to \mathbb{C}$$
$$G(W) = \mathrm{State} \stackrel{ne}{\to} W \stackrel{mon,ne}{\to} \mathrm{UPred}(HS)$$
$$G(g) = \lambda H.\, \lambda st.\, \lambda x.\, H(st)(g(x))$$

We first show that K, R, and G are well-defined mappings.

Lemma 3 (World finite partial mapping). *For all f and ϕ, $K(f)(\phi)$ is a finite partial mapping.* ■

Lemma 4 (Heap segment predicate monotone). *For all g, H, and st*

$$G(g)(H)(st)$$

is non-expansive. ■

Lemma 5 (Heap segment predicate non-expansive). *For all g, H, and st*

$$G(g)(H)(st)$$

is monotone. ■

Next we show that K, R, and G are in fact functors:

Lemma 6 (K functorial).

1. *$K(f) : K(X) \to K(Y)$ is monotone and non-expansive for $f : X \stackrel{mon,ne}{\to} Y$*
2. *$K(f \circ g) = K(f) \circ K(g)$ for $f : Z \stackrel{ne}{\to} Y$ and $g : X \stackrel{ne}{\to} Z$*
3. *$K(id) = id$*

$\blacksquare$

Lemma 7 (R functorial).

1. *$R(f) : R(X) \to R(Y)$ is non-expansive and monotone for $f : X \xrightarrow{ne} Y$*
2. *$R(f \circ g) = R(f) \circ R(g)$ for $f : Z \xrightarrow{ne} Y$ and $g : X \xrightarrow{ne} Z$*
3. *$R(id) = id$*

$\blacksquare$

Lemma 8 (G functorial).

1. *$G(f) : G(Y) \to G(X)$ is non-expansive for $f : X \xrightarrow{mon,ne} Y$*
2. *$G(f \circ g) = G(g) \circ G(f)$ for $f : Z \xrightarrow{ne} Y$ and $g : Y \xrightarrow{ne} Z$*
3. *$G(id) = id$*

$\blacksquare$

We now compose the above functors into the functor we actually want to use: $F = K \circ R \circ G$, $F : \mathbb{P}^{op} \to \mathbb{P}$.

Lemma 9 (F functorial).

1. *$F(f) : F(Y) \to F(X)$ is monotone and non-expansive for $f : X \xrightarrow{mon,ne} Y$*
2. *$F(f \circ g) = F(g) \circ F(f)$ for $f : Z \xrightarrow{ne} Y$ and $g : Y \xrightarrow{ne} Z$*
3. *$F(id) = id$*

$\blacksquare$

Lemma 10 (F locally non-expansive). *For all $f, g : X \to Y$, if $f \stackrel{n}{=} g$, then $F(f) \stackrel{n}{=} F(g)$.* $\blacksquare$

With F being locally-non-expansive, we can pre- or post-compose with later ($\blacktriangleright$) to get a locally contractive function. In this case we construct F' by post-copmosition of $\blacktriangleright$:

$$F'(\text{Wor}) = \blacktriangleright(F(\text{Wor}))$$

We have a theorem that gives us a solution to the recurisve domain equation

$$\text{Wor} \cong F'(\text{Wor}) = \blacktriangleright(\mathbb{N} \xrightarrow{fin} (\text{State} \times \text{Rel} \times (\text{State} \to \text{Wor} \xrightarrow{mon,ne} \text{UPred}(\text{HeapSegment}))))$$

The solution to the recursice domain equations is presented by Birkedal et al. [2010]. They solve it in pre-ordered, non-empty, complete, 1-bounded ultrametric spaces, but they have a simple correspondence to pre-ordered c.o.f.e.'s.

2.2 Worlds

Assume preordered c.o.f.e. Wor and isomorphism ξ such that:

$\xi : \text{Wor} \cong \blacktriangleright(\mathbb{N} \xrightarrow{fin} (\text{State} \times \text{Rel} \times (\text{State} \overset{ne}{\to} (\text{Wor} \xrightarrow{mon, ne} \text{UPred}(\text{HeapSegment})))))$

ξ is a morph. in $\mathbb{P}$, so mono + n.e. lemmas.

We now define regions as

$$\text{Region} \stackrel{def}{=} (\text{State} \times \text{Rel} \times (\text{State} \overset{ne}{\to} (\text{Wor} \xrightarrow{mon, ne} \text{UPred}(\text{HeapSegment}))))$$

define region names to be natural numbers, i.e.,

$$\text{RegionName} \stackrel{def}{=} \mathbb{N}$$

and define worlds as

$$\text{World} \stackrel{def}{=} \text{RegionName} \xrightarrow{fin} \text{Region}$$

To define future worlds and regions, We use the ordering inherited from the preordered c.o.f.e.'s.

Definition 8 (Future worlds). *For $W, W' \in$ World*

$$W' \sqsupseteq W \quad \textit{iff} \quad \begin{array}{c} \text{dom}(W') \supseteq \text{dom}(W) \\ \textit{and} \\ \forall r \in \text{dom}(W).\, W'(r) \sqsupseteq W(r) \end{array}$$

■

Definition 9 (Future regions). *For regions $(s_2, \phi_2, H_2), (s_1, \phi_1, H_1) \in$ Region*

$$(s_2, \phi_2, H_2) \sqsupseteq (s_1, \phi_1, H_1) \quad \textit{iff} \quad (\phi_1, H_1) = (\phi_2, H_2) \textit{ and } (s_1, s_2) \in \phi_2$$

■

Definition 10 (n-subset for regions). *For regions $(s_1, \phi_1, H_1), (s_2, \phi_2, H_2) \in$ Region*

$$(s_1, \phi_1, H_1) \overset{n}{\subseteq} (s_2, \phi_2, H_2) \quad \textit{iff} \quad \begin{array}{c} (s_1, \phi_1) = (s_2, \phi_2) \\ \textit{and} \\ \forall W \in \text{Wor}.\, H_1\ s_1\ W \overset{n}{\subseteq} H_2\ s_2\ W \end{array}$$

■

Definition 11 (Heap satisfaction/erasure).

Parameterize?

$$\begin{array}{l} hs :_n W \\ \quad \textit{iff} \\ \exists R : \text{dom}(W) \to \text{HeapSegment}. \\ \quad hs = \biguplus_{r \in \text{dom}(W)} R(r) \\ \quad\quad \textit{and} \\ \quad \forall r \in \text{dom}(W).\, \forall n' < n.\, (n', R(r)) \in W(r).H(W(r).s)(\xi^{-1}(W)) \end{array}$$

■

Lemmas about heap satisfaction

a) <u>Lemma</u> downwards closed

For all hs, n, n', W

$n' \leq n$ and $hs :_n W$

$\Rightarrow hs :_{n'} W$

b) <u>Lemma</u> non-expansive

For all hs, n, W, W'

$W \stackrel{n}{=} W'$ and $hs :_n W$

$\Rightarrow hs :_n W'$

Lemma a) downwards closed

assume $n' \leq n^{(I)}$ and $hs :_n W^{(II)}$

show $hs :_{n'} W$

From (II) get $R : dom(W) \to HeapSegment$ s.t.

$$hs = \biguplus_{r \in dom(W)} R(r) \quad ^{(III)}$$

and

$$\forall r \in dom(W), n'' < n. \quad (n'', R(r)) \in W(r).\ H(W(r).s)(\xi^{-1}(W)) \quad ^{(IV)}$$

To show $hs :_{n'} W$ pick R. The first condition follows from (III). The second condition is

$$\forall r \in dom(W), n'' < n'. \quad (n'', R(r)) \in W(r).\ H(W(r).s)(\xi^{-1}(W))$$

let $r \in dom(W)$ and $n'' < n'$ be given. As we have $n' < n$, we get $n'' < n$, so IV can be used to get the desired result.

p. 17 2

Lemma b) non-expansive.

Assume $W \stackrel{n}{=} W'$ $^{(I)}$ and $hs :_n W$ $^{(II)}$

Show $hs :_n W'$. $^{(III)}$

From (II) get R. Use the same R to show (III), we use that (I) gives us $dom(W) = dom(W')$. It follows then from (II) that

$$hs = \biguplus_{\substack{r \in dom(W') \\ \shortparallel \\ dom(W)}} R(r)$$

To show the second condition, let $r \in dom(W)$ and $n' < n$ be given.

Use (II) to conclude

$$(n', R(r)) \in W(r).H(W(r).s)(\xi^{-1}(W)) \quad (IV)$$

From (I) we get $W(r).H \stackrel{n}{=} W'(r).H$,

$W(r).s = W'(r).s$
(n-equality for State is equality.)

$$W(r).H(W(r).s) \stackrel{n}{=} W'(r).H(W'(r).s)$$

next: World $\to$ World

As $W \stackrel{n}{=} W'$, we have $\text{next } W \stackrel{n+1}{=} \text{next } W'$, so

$$\xi^{-1}(\text{next } W) \stackrel{n+1}{=} \xi^{-1}(\text{next } W')$$

which implies n-equality (usually we leave out the next).

as $W(r).H$ is non-expansive, then

$$W(r).H(W(r).s)(\xi^{-1} W) \stackrel{n}{=} W(r).H(W(r).s)(\xi^{-1} W')$$

Which w/ (IV) gives the desired result. P. 17. 3

2.3 Logical Relation

Our logical relation is defined using multiple recursive definitions, so the definitions in the following subsections are defined simultaneously. We want to define the value relation as the fixed-point given by Banach's fixed-point theorem, so all our definitions will be parameterized with the value relation.

2.3.1 Observation Relation

In order to define the expression relation, we define an observation relation.

$$
\begin{aligned}
\mathcal{O} \quad &: \quad \text{World} \xrightarrow{ne} \text{UPred}(\text{Reg} \times \text{HeapSegment}) \\
\mathcal{O}(W) &\stackrel{def}{=} \{(n, (reg, hs)) \mid \\
&\qquad (\forall heap_f, heap', i \leq n.\, (reg, hs \uplus heap_f) \rightarrow_i (halted, heap') \\
&\qquad\quad \Rightarrow \exists W' \sqsupseteq W.\, \exists hs'.\, heap' = hs' \uplus heap_f \wedge hs' :_{n-i} W'
\end{aligned}
$$

A pair of a register and a heap segment is "good" if we can put it together with a frame heap, so we can execute it. The execution should then end up in a heap where the frame remains the same and the remaining heap segment satisfies the world.

Note that the operational semantic is total, so we cannot get stuck. If the execution ends up in a *failed* configuration, then we do not care about the heap and the registers. This is why, we only have requirements on the result when we end up in a *halted* configuration.

The following lemmas show that the observation relation is well-defined.

Lemma 11 (Observation relation uniformity).

$$
\begin{aligned}
&\forall n' < n.\, \forall W.\, \forall reg.\, \forall hs. \\
&\quad (n, (reg, hs)) \in \mathcal{O}(W) \Rightarrow (n', (reg, hs)) \in \mathcal{O}(W)
\end{aligned}
$$

√ 5.36

■

Lemma 12 (Observation relation non-expansive in worlds).

HW

$$
\begin{aligned}
&\forall W, W', n. \\
&\quad W \stackrel{n}{=} W' \Rightarrow \mathcal{O}(W) \stackrel{n}{=} \mathcal{O}(W')
\end{aligned}
$$

■

2.3.2 Regiser-File Relation

This relation is used in the definition of the continuation relation as well as the expression relation.

$$
\begin{aligned}
\mathcal{R} \quad &: \quad (\text{World} \xrightarrow{mon, ne} \text{UPred}(\text{Word})) \xrightarrow{ne} \text{World} \xrightarrow{mon, ne} \text{UPred}(\text{Reg}) \\
\mathcal{R} &\stackrel{def}{=} \lambda\mathcal{V}.\, \lambda W.\, \{(n, reg) \mid \forall r \in \text{RegisterName} \setminus \{\text{pc}\}. \\
&\qquad\qquad (n, reg(r)) \in \mathcal{V}(W)\}
\end{aligned}
$$

Well-formedness lemmas for this definition:

Lemma 12

Assume $W_1 \overset{n}{=} W_2^{\sharp}$ (II)

Show $\mathcal{O}(W_1) \overset{n}{=} \mathcal{O}(W_2^{\sharp})$

Assume $(k,(reg,hs)) \in \mathcal{O}(W_1)$ (I) where $k<n$

let $heap_f$, $heap'$, $i \leq k$ be given and

assume $(reg, hs \uplus heap_f) \to_i (halted, heap')$

By assumption (I) there exists $W_1' \sqsupseteq W_1$ (III) and hs' st.

$heap' = hs' \uplus heap_f \wedge hs' :_{k-i} W_1'$

By lemma ...? (II) and (III) gives $W_2' \sqsupseteq W_2$ s.t. $W_2' \overset{n}{=} W_1'$.

using hs' and $W_2^{\sharp}$ we have $heap' = hs' \uplus heap_f$ and by lemma we get $hs' :_{k-i} W_2'$.

Lemma heap sat. ne. (already in doc.)

$hs :_n W$ & $W \overset{n}{=} W' \Rightarrow hs :_n W'$

for p. 18

Lemma 13 (Register relation uniformity).

$$\begin{aligned}&\forall \mathcal{V}, n' \leq n. \forall W. \forall reg.\\&\quad (n, reg) \in \mathcal{R}(\mathcal{V})(W) \Rightarrow (n', reg) \in \mathcal{R}(\mathcal{V})(W)\end{aligned}$$

– HW

■

Lemma 14 (Register relation montone in worlds).

$$\begin{aligned}&\forall \mathcal{V}, n. \forall W' \sqsupseteq W. \forall reg.\\&\quad (n, reg) \in \mathcal{R}(\mathcal{V})(W) \Rightarrow (n, reg) \in \mathcal{R}(\mathcal{V})(W')\end{aligned}$$

HW

■

Lemma 15 (Register relation non-expansive in value relation).

$$\forall \mathcal{V}, \mathcal{V}', n. \mathcal{V} \stackrel{n}{=} \mathcal{V}' \Rightarrow \mathcal{R}(\mathcal{V}) \stackrel{n}{=} \mathcal{R}(\mathcal{V}')$$

HW

■

R downwards closed, see HW pages. (7)

2.3.3 Continuation Relation

The continuation relation is used in the definition of the expression relation. The continuation relation ensures that if you continue execution through a continuation, then it will result in a good result according to the world.

$$\mathcal{K} \quad : \quad (\text{World} \xrightarrow{\text{mon}, \text{ne}} \text{UPred}(\text{Word})) \xrightarrow{\text{ne}} \text{World} \xrightarrow{\text{mon}, \text{ne}} \text{UPred}(\text{Word})$$

$$\begin{aligned}\mathcal{K} \stackrel{def}{=} \lambda \mathcal{V}. \lambda W. \{(n, c) \mid \; & (n, c) \in \mathcal{V}(W) \wedge \\ & \forall W' \sqsupseteq W, n' < n. \forall hs :_{n'} W'. \forall reg, (n', reg) \in \mathcal{R}(\mathcal{V})(W'). \\ & \quad (n', (reg[\text{pc} \mapsto updatePcPerm(c)], hs)) \in \mathcal{O}(W')\}\end{aligned}$$

Well-definedness lemmas:

Lemma 16 (Continuation relation uniformity).

$$\begin{aligned}&\forall \mathcal{V}. \forall n' < n. \forall W. \forall c.\\&\quad (n, c) \in \mathcal{K}(\mathcal{V})(W) \Rightarrow (n', c) \in \mathcal{K}(\mathcal{V})(W)\end{aligned}$$

HW

■

Lemma 17 (Continuation relation monotone in worlds).

$$\begin{aligned}&\forall \mathcal{V}. \forall n. \forall W' \sqsupseteq W. \forall c.\\&\quad (n, c) \in \mathcal{K}(\mathcal{V})(W) \Rightarrow (n, c) \in \mathcal{K}(\mathcal{V})(W')\end{aligned}$$

HW

■

Lemma 18 (Continuation relation non-expansive in value relation).

$$\forall \mathcal{V}, \mathcal{V}', n. \mathcal{V} \stackrel{n}{=} \mathcal{V}' \Rightarrow \mathcal{K}(\mathcal{V}) \stackrel{n}{=} \mathcal{K}(\mathcal{V}')$$

HW

■

K downwards closed missing see HW pages (8)

Lemma 13 proof

Let $\mathcal{V}$, $n' \leq n$, W, and reg be given

assume $(n, reg) \in \mathcal{R}(\mathcal{V})(W)^{(I)}$

show $(n', reg) \in \mathcal{R}(\mathcal{V})(W)$

if $n' = n$, done

Let $r \in RegName \setminus \{pc\}$ be given. show

$(n', reg(r)) \in \mathcal{V}(W)$

By assump. (I)

$(n, reg(r)) \in \underline{\mathcal{V}(W)}$

Uniform pred. on words, so for all

$k \leq n$ $(k, reg(r)) \in \mathcal{V}(W)$

Result follows from $n' < n$.

Proof lemma 14

Assume $W_2 \sqsupseteq W_1$ $^{(\mathrm{I})}$ and $(n, reg) \in \mathcal{R}(\mathcal{V})(W_1)$ $^{(\mathrm{I})}$

Show $(n, reg) \in \mathcal{R}(\mathcal{V})(W_2)$

let $r \in RN \setminus \{pc\}$ be given

(I) gives $(n, reg(r)) \in \mathcal{V}(W_1)$

$\mathcal{V}$ mono, so by (I)

$(n, reg(r)) \in \mathcal{V}(W_2)$.

Lemma 15

Assume $\mathcal{V} \stackrel{n}{=} \mathcal{V}'$

Let W be given, show

$$R(\mathcal{V})(W) \stackrel{n}{=} R(\mathcal{V}')(W)$$

let $(k, reg) \in$

let $r \in RN\cancel{X}Spc$ be given

$$(k, reg(r)) \in \mathcal{V}'(W)$$

By def of n-equal

$$\forall W.\ \mathcal{V}(W) \stackrel{n}{=} \mathcal{V}'(W)$$

so $(k, reg(r)) \in \mathcal{V}'(W)$.

p. 19 3

Lemma 16

Assume $n' < n$ and $(n,c) \in \mathcal{K}(\mathcal{V})(W)^{(I)}$

Show $(n',c) \in \mathcal{R}(\mathcal{V})(W)$

By assumption $(n,c) \in \underline{\mathcal{V}(W)}$ ∈ UPred

$(n',c) \in \mathcal{V}(W)$

Now let $W' \sqsupseteq W$ be given and $n'' < n'$ and hs s.t. $hs :_{n''} W'$ and reg s.t. $(n'', reg) \in \mathcal{R}(\mathcal{V})(W')$

By assumption ~~and downwards closure of heap satisfaction~~ $n'' < n$ get

$(n'', (reg[pc \mapsto updPcA(c)], hs) \in \mathcal{O}(W')$

whic is what we needed.

P19 4

Lemma 17

Assume $W_2 \sqsupseteq W_1$ and $(n,c) \in K(\mathcal{V})(W_1)$ (I)

Show $(n,c) \in K(\mathcal{V})(W_2)$

- show $(n,c) \in \mathcal{V}(W_2)$ follows from $\mathcal{V}$ mono + $W_2 \sqsupseteq W_1$
- let $W_2' \sqsupseteq W_2$, $n' < n$, hs and reg be given s.t. $hs :_{n'} W_2'$ and $(n', reg) \in R(\mathcal{V})(W_2')$

As $W_2 \sqsupseteq W_1$, we have $W_2' \sqsupseteq W_1$ by trans.
The result now follows by assump. (I).

Lemma 18

Assume $\mathcal{V} \stackrel{n}{=} \mathcal{V}'$
show $\mathcal{K}(\mathcal{V}) \stackrel{n}{=} \mathcal{K}(\mathcal{V}')$

Amounts to: let W be given, show
$\mathcal{K}(\mathcal{V})(W) \stackrel{n}{=} \mathcal{K}(\mathcal{V}')(W)$

let $(k, c) \in \mathcal{K}(\mathcal{V})(W)^{(I)}$ for $k < n$

- show $(k, c) \in \mathcal{V}(W)$
  by def of n-eq. $(k, c) \in \mathcal{V}'(W)$ for all W, so in particular for our W.

- Let $W' \sqsupseteq W$, $k' < k$, $hs :_{n'} W'$, $(k', reg) \in \mathcal{R}(\mathcal{V})(W)$
  be given
  By lemma 15, Reg. rel. ne, $(k', reg) \in \mathcal{R}(\mathcal{V})(W)$
  So our assumption$^{(I)}$ gives the result, i.e.
  $(n', (reg[pc \mapsto upp(c)], hs)) \in O(W')$

p. 19 6

Lemma R downwards closed

for all $\mathcal{V}, W, n, n'$ $\quad(*)$

if $n' \leq n$ and $n \in R(\mathcal{V})(W)$,

then $n' \in R(\mathcal{V})(W)$

Proof

Assume $n' \leq n$ and $n \in R(\mathcal{V})(W)$.

Show $n' \in R(\mathcal{V})(W)$

let $r \in RegName \setminus \{pc\}$ be given.

From (I), we get

$(n, reg(r)) \in \mathcal{V}(W)$

As $\mathcal{V}(W)$ is downwardsclosed and $n' \leq n$, we get

$(n', reg(r)) \in \mathcal{V}(W)$.

Lemma $\mathcal{R}$ is downwards closed

For all $\mathcal{V}, W, n, n', c$

if $n' \leq n$ and $(n, c) \in \mathcal{R}(\mathcal{V})(W)$, (I)

then $(n', c) \in \mathcal{R}(\mathcal{V})(W)$

Proof

Assume $n' \leq n$ and $(n, c) \in \mathcal{R}(\mathcal{V})(W)$

Show $(n', c) \in \mathcal{R}(\mathcal{V})(W)$

As $\mathcal{V}(W)$ is downwards closed, and by assumption (I), we have $(n, c) \in \mathcal{V}(W)$ it follows that

$(n', c) \in \mathcal{V}(W)$.

Now let $W' \sqsupseteq W$, $n'' < n'$, $hs :_{n''} W'$ and $(n'', reg) \in \mathcal{R}(\mathcal{V})(W)$ be given. Use (I) (as $n'' < n' < n$) to get

$(n'', (reg[pc \mapsto uPP(c)], hs)) \in \mathcal{O}(W)$

Which is what we needed to show.

2.3.4 Expression Relation

The expression relation is defined as follows:

$$\mathcal{E} \quad : \quad (\text{World} \xrightarrow{\text{mon},\text{ne}} \text{UPred}(\text{Word})) \xrightarrow{\text{ne}} \text{World} \xrightarrow{\text{ne}} \text{UPred}(\text{Word})$$

$$\begin{aligned}\mathcal{E} \stackrel{def}{=} \lambda \mathcal{V}.\, \lambda W.\, \{(n, pc) \mid\ & \forall n' \leq n. \\ & \forall (n', reg) \in \mathcal{R}(\mathcal{V})(W). \\ & \forall (n', c) \in \mathcal{K}(\mathcal{V})(W). \\ & \forall hs :_{n'} W. \\ & (n', (reg[r_0 \mapsto c][\text{pc} \mapsto pc], hs)) \in \mathcal{O}(W)\}\end{aligned}$$

Well-definedness lemmas:

Lemma 19 (Expression relation uniformity). — HW

$$\forall \mathcal{V}. \forall n' \leq n. \forall W. \forall pc. \\ (n, pc) \in \mathcal{E}(\mathcal{V})(W) \Rightarrow (n', pc) \in \mathcal{E}(\mathcal{V})(W)$$

∎

Lemma 20 (Expression relation non-expansive in world). HW

$$\forall \mathcal{V}. \forall W_1 \stackrel{n}{=} W_2.\, \mathcal{E}(\mathcal{V})(W_1) \stackrel{n}{=} \mathcal{E}(\mathcal{V})(W_2)$$

∎

Lemma 21 (Expression relation non-expansive in value relation). HW

$$\forall \mathcal{V}, \mathcal{V}', n.\, \mathcal{V} \stackrel{n}{=} \mathcal{V}' \Rightarrow \mathcal{E}(\mathcal{V}) \stackrel{n}{=} \mathcal{E}(\mathcal{V}')$$

∎ $\mathcal{E}$ downwards closed missing see HW notes (4)

2.3.5 Standard Region

The following standard region is used in the definition of the value relation. Specifically, it is used in the *readCondition* and the *readWriteCondition* (to be defined next)

$$\iota_{start,end} : (\text{World} \xrightarrow{\text{mon},\text{ne}} \text{UPred}(\text{Word})) \xrightarrow{\text{ne}} \text{Region}$$

$$\iota_{base,end} \stackrel{def}{=} \lambda\, \mathcal{V}.\, ((base, end), =, H_{std}(\mathcal{V}))$$

Wor —

$$H_{std} : (\text{World} \xrightarrow{\text{mon},\text{ne}} \text{UPred}(\text{Word})) \xrightarrow{\text{ne}} \text{State} \xrightarrow{\text{ne}} \cancel{\text{World}} \xrightarrow{\text{mon},\text{ne}} \text{UPred}(\text{HeapSegment})$$

$$H_{std}\ \mathcal{V}\ (base, end)\ \hat{W} \stackrel{def}{=} \left\{ (n, hs) \,\middle|\, \begin{array}{l} \text{dom}(hs) = [base, end] \wedge \\ \forall a \in [base, end].\, (n-1, hs(a)) \in \mathcal{V}(\xi\, \hat{W}) \end{array} \right\}$$ — — ?

Lemma 19

let $n' < n$ be given

assume $(n, pc) \in \mathcal{E}(\mathcal{V})(W)$

show $(n', pc) \in \mathcal{E}(\mathcal{V})(W)$

let $n'' \leq n'$ be given (and so on)

Since $n'' \leq n$, we can use the assumption to get the desired result.

~~Lemma 20~~ $\mathcal{E}$ n.e. in Worlds.

let $W_1 \overset{n}{=} W_2$

assume $(k, pc) \in \mathcal{E}(\mathcal{V})(W_1)^{(I)}$ for $k < n$.

show $(k, pc) \in \mathcal{E}(\mathcal{V})(W_2)$

let $k' \leq k$, $(k', reg) \in \mathcal{R}(\mathcal{V})(W_2)$, $(k', c) \in \mathcal{K}(\mathcal{V})(W_2)$,

$hs :_{k'} W_2$ be given.

By $\mathcal{R}$ n.e in W get $(k', reg) \in \mathcal{R}(\mathcal{V})(W_1)$

By $\mathcal{K}$ n.e in W get $(k', c) \in \mathcal{R}(\mathcal{V})(W_1)$

By heap. sat. n.e. in W get $hs :_{k'} W_1$

By (I), we now get

$(k'', (reg[r_0 \mapsto c][pc \mapsto pc], hs)) \in \mathcal{O}(W_1)$

By $\mathcal{O}$ rel. n.e in worlds get

$\mathcal{O}(W_1) \overset{n}{=} \mathcal{O}(W_2)$ and as $k'' < n$ get

$(k'', (reg[r_0 \mapsto c][pc \mapsto pc], hs)) \in \mathcal{O}(W_2)$

Lemma 21

$\mathcal{E}$ n.e. in $\mathcal{V}$.

assume $\mathcal{V} \overset{n}{\sqsupseteq} \mathcal{V}'$ (I) show $\mathcal{E}(\mathcal{V}) \overset{n}{\sqsupseteq} \mathcal{E}(\mathcal{V}')$

To this end let W be given and show

$$\mathcal{E}(\mathcal{V})(W) \overset{n}{\sqsupseteq} \mathcal{E}(\mathcal{V}')(W).$$

let $(k, pc) \in$ for $k < n$.

show $(k, pc) \in \mathcal{E}(\mathcal{V}')(W)$

To this end let $k' \leq k$, $(k', reg) \in \mathcal{R}(\mathcal{V})(W)$, $(k', c) \in \mathcal{K}(\mathcal{V}')(W)$,

$hs :_{k'} W$ be given.

By $\mathcal{R}$ and $\mathcal{K}$ being n.e. in $\mathcal{V}$, and (I),

we get $(k', reg) \in \mathcal{R}(\mathcal{V})(W)$ and $(k', c) \in \mathcal{K}(\mathcal{V})(W)$

Now by assumption $(k, pc) \in \mathcal{E}(\mathcal{V})(W)$, get

$(k', (reg[r_0 \mapsto c][pc \mapsto pc])) \in \mathcal{O}(W)$

Which is what we needed.

p. 20 3

Lemma $\mathcal{E}$ downwards closed

For all $\mathcal{V}, W, pc, n', n$ (I)

if $n' \leq n$ and $(n, pc) \in \mathcal{E}(\mathcal{V})(W)$,

then $(n', pc) \in \mathcal{E}(\mathcal{V})(W)$

Assume $n' \leq n$ and $(n, pc) \in \mathcal{E}(\mathcal{V})(W)$

Show $(n', pc) \in \mathcal{E}(\mathcal{V})(W)$

To this end let n'', reg, c, hs be given s.t. $(n'', c) \in \mathcal{K}(\mathcal{V})(W)$, and $(n'', reg) \in \mathcal{R}(\mathcal{V})(W)$, $n'' \leq n'$,

$hs :_{n''} W$.

By (I), as $n'' \leq n' \leq n$, we get

$(n'' (reg[r_0 \mapsto c][\underline{pc} \mapsto pc], hs)) \in \mathcal{O}(W)$

register name

Which is exactly what we needed.

As mentioned previously, the set of states contains the "necessary" states. For the above to make sense, the set of states contains pairs of natural numbers $(base, end)$.

The well-definedness lemmas for the above is:

Lemma 22 (H_{std} is monotone in the worlds).

HW

$$\forall \mathcal{V}.\, \forall base, end.\, \forall W' \sqsupseteq W.$$
$$H_{std}\ \mathcal{V}\ (base, end)\ W' \supseteq H_{std}\ \mathcal{V}\ (base, end)\ W$$

∎

Lemma 23 (H_{std} is non-expansive in the worlds).

HW

$$\forall \mathcal{V}.\, \forall base, end.\, \forall n.\, \forall W_1 \stackrel{n}{=} W_2.$$
$$H_{std}\ \mathcal{V}\ (base, end)\ W_1 \stackrel{n}{=} H_{std}\ \mathcal{V}\ (base, end)\ W$$

∎

Lemma 24 (H_{std} is non-expansive in the value relation).

HW

$$\forall \mathcal{V}, \mathcal{V}'.\, \forall n.\, \mathcal{V} \stackrel{n}{=} \mathcal{V}' \Rightarrow$$
$$H_{std}\ \mathcal{V} \stackrel{n}{=} H_{std}\ \mathcal{V}'$$

∎

Lemma 25 ($\iota_{base,end}$ is non-expansive in the value relation).

$$\forall base, end.\, \forall \mathcal{V}, \mathcal{V}'.\, \forall n.\, \mathcal{V} \stackrel{n}{=} \mathcal{V}' \Rightarrow$$
$$\iota_{base,end}\ \mathcal{V} \stackrel{n}{=} \iota_{base,end}\ \mathcal{V}'$$

HW

∎

Missing
- Hstd downwardsclosed (5)
- Hstd non-expansive in State (6)

Lemma 22

H_{std} mono in World:

let $\mathcal{V}$, b, e, $W' \sqsupseteq W$ be given.

Show $H_{std}\, \mathcal{V}\, (b,e)\, W' \supseteq H_{std}\, \mathcal{V}\, (b,e)\, W$

assume

$(n, hs) \in H_{std}\, \mathcal{V}\, (b,e)\, W$ (I)

show

1) $dom(hs) = [b,e]$

2) $\forall a \in [b,e].\ (n-1, hs(a)) \in \mathcal{V}(\xi\, W')$

1) follows directly from first condition in (I).

2) let $a \in [b,e)$ be given s.t.

$(n-1, hs(a)) \in \mathcal{V}(\xi\, W)$

$\mathcal{V}$ is monotone, and ξ monotone.

Lemma $W' \sqsupseteq W$

$\Rightarrow \xi\, W' \sqsupseteq \xi\, W$ — proof by construction of ξ, morphism in $\mathbb{P}$.

P. 21 1

lemma 23 $\quad H_{std}$ n.e. in worlds

let $W_1 \overset{n}{=} W_2$

Show $H_{std}\ \mathcal{V}\ (b,e)\ W_1 \overset{n}{=} H_{std}\ \mathcal{V}\ (b,e)\ W_2$

let $k<n$ and
$(k,hs) \in H_{std}\ \mathcal{V}\ (b,e)\ W_1$ (I)

Show

- $dom(hs) = [b,e]$, follows directly from (I)
- $\forall a \in [b,e].\ (k-1, hs(a)) \in \mathcal{V}(\xi\, W_2)$

  let a be given. By (I):
  $(k-1, hs(a)) \in \mathcal{V}(\xi\, W_1)$ (II)

  ξ n.e. so $\xi\, W_1 \overset{n}{=}_{\triangleright World} \xi\, W_2$ (as later worlds, i.e. $\triangleright$World)

  So we have $\xi\, W_1 \overset{n-1}{=}_{World} \xi\, W_2$

  As $\mathcal{V}$ is n.e., we have
  $\mathcal{V}(\xi\, W_1) \overset{n-1}{=} \mathcal{V}(\xi\, W_2)$

  As $k<n$, we also have $k-1 < n-1$, so using (II), we get the desired result.

p. 21 $\quad$ 2

Lemma 24

H_{std} ne in $\mathcal{V}$

Assume $\mathcal{V} \overset{n}{\doteq} \mathcal{V}'$

Show $H_{std}\,\mathcal{V} \overset{n}{\doteq} H_{std}\,\mathcal{V}'$ to this end let

(b,e) and W be given and show

$$H_{std}\mathcal{V}(b,e)W \overset{n}{\doteq} H_{std}\,\mathcal{V}'(b,e)\,W$$

Let $(k,hs) \in H_{std}\ \mathcal{V}\ (b,e)\ W$ (I) for some hs and $k<n$.

Show

- $dom(hs) = [b,e]$, follows directly from (I)
- $\forall a \in [b,e].\ \ (k-1, hs(a)) \in \mathcal{V}'(\xi\ W)$

Let $a \in [b,e]$ given. From (I) get

$$(k-1, hs(a)) \in \mathcal{V}(\xi\ W) \quad \text{(II)}$$

by $\mathcal{V} \overset{n}{\doteq} \mathcal{V}'$ so $\mathcal{V}(\xi\ W) \overset{n}{\doteq} \mathcal{V}'(\xi\ W)$

from this and (II) conclude

$$(k-1, hs(a)) \in \mathcal{V}'(\xi\ W).$$

p21 3

Lemma 25

Assume $V \overset{n}{=} V'$

show

$L_{b,e} V \overset{n}{=} L_{b,e} V'$

i.e.

- $(b,e) \overset{n}{=} (b,e)$: trivial, refl.
- "=" $\overset{n}{=}$ "=" : trivial, refl
- $H_{std} V \overset{n}{=} H_{std} V'$: lemma 24, H_{std} ne in V_{rel}.

p. 21 4

Lemma H_{std} downwardsclosed

For all $\mathcal{V}, \overset{(b,e)}{\underset{s}{\shortparallel}}, \hat{W}, n', n, hs$

if $n' \leq n$ and $(n, hs) \in H_{std}\ \mathcal{V}\ s\ \hat{W}$,

then $(n', hs) \in H_{std}\ \mathcal{V}\ s\ \hat{W}$

Proof

Assume $n' \leq n$ and $(n, hs) \in H_{std}\ \mathcal{V}\ s\ \hat{W}$ (I)

Show $H_{std}\ \mathcal{V}\ s\ \hat{W}$, i.e

- $dom(hs) = [b, e]$, follows directly from (I).
- $\forall a \in [b, e].\ (n'-1, hs(a)) \in \mathcal{V}(\xi(\hat{W}))$

Let a be given. By (I) get

$$(n-1, hs(a)) \in \mathcal{V}(\xi(\hat{W}))$$

As $\mathcal{V}$ is downwards closed in the world and $n' \leq n \Rightarrow n'-1 \leq n-1$, it must be the case that

$$(n'-1, hs(a)) \in \mathcal{V}(\xi(\hat{W})).$$

p. 21 (5)

Lemma H_{std} non-expansive in state.

forall $\mathcal{V}$, (b,e), (b',e')
if $(b,e) \overset{n}{=} (b',e')$, then
$H_{std}\ \mathcal{V}\ (b,e) \overset{n}{=} H_{std}\ \mathcal{V}\ (b',e')$

Proof (trivial)

Assume $(b,e) \overset{n}{=} (b',e') \Rightarrow (b,e) = (b',e')$

$H_{std}\ \mathcal{V}\ (b,e) = H_{std}\ \mathcal{V}\ (b',e')$

2.3.6 Capability Conditions

The definition of the value relation has the same conditions several times, so to define it consisely, we define the following conditions.

$$
\begin{aligned}
\mathit{readCondition} &: (\mathrm{World} \overset{mon,ne}{\rightarrow} \mathrm{UPred}(\mathrm{Word})) \overset{ne}{\rightarrow} (\mathrm{Addr}^2 \times \mathrm{World}) \overset{mon,ne}{\rightarrow} \mathrm{P}^{\downarrow}(\mathbb{N}) \\
\mathit{readCondition}(\mathcal{V})(\mathit{base}, \mathit{end}, W) &= \{n \mid \exists r \in \mathrm{RegionName}. \\
&\qquad \exists[\mathit{base}', \mathit{end}'] \supseteq [\mathit{base}, \mathit{end}]. \\
&\qquad\quad W(r) \overset{n-1}{\subseteq} \iota_{\mathit{base}',\mathit{end}'}(\mathcal{V})\}
\end{aligned}
$$

$$
\begin{aligned}
\mathit{readWriteCondition} &: (\mathrm{World} \overset{mon,ne}{\rightarrow} \mathrm{UPred}(\mathrm{Word})) \overset{ne}{\rightarrow} (\mathrm{Addr}^2 \times \mathrm{World}) \overset{mon,ne}{\rightarrow} \mathrm{P}^{\downarrow}(\mathbb{N}) \\
\mathit{readWriteCondition}(\mathcal{V})(\mathit{base}, \mathit{end}, W) &= \{n \mid \exists r \in \mathrm{RegionName}. \\
&\qquad \exists[\mathit{base}', \mathit{end}'] \supseteq [\mathit{base}, \mathit{end}]. \\
&\qquad\quad W(r) \overset{n-1}{=} \iota_{\mathit{base}',\mathit{end}'}(\mathcal{V})\}
\end{aligned}
$$

$$
\begin{aligned}
\mathit{executeCondition} &: (\mathrm{World} \overset{mon,ne}{\rightarrow} \mathrm{UPred}(\mathrm{Word})) \overset{ne}{\rightarrow} (\mathrm{Addr}^2 \times \mathrm{Perm} \times \mathrm{World}) \overset{mon,ne}{\rightarrow} \mathrm{P}^{\downarrow}(\mathrm{I} \\
\mathit{executeCondition}(\mathcal{V})(\mathit{base}, \mathit{end}, \mathit{perm}, W) &= \{n \mid \forall n' < n. \forall W' \sqsupseteq W. \\
&\qquad \forall a \in [\mathit{base}, \mathit{end}]. \\
&\qquad\quad (n', (\mathit{perm}, \mathit{base}, \mathit{end}, a)) \in \mathcal{E}(\mathcal{V})(W')\}
\end{aligned}
$$

$$
\begin{aligned}
\mathit{entryCondition} &: (\mathrm{World} \overset{mon,ne}{\rightarrow} \mathrm{UPred}(\mathrm{Word})) \overset{ne}{\rightarrow} (\mathrm{Addr}^3 \times \mathrm{World}) \overset{mon,ne}{\rightarrow} \mathrm{P}^{\downarrow}(\mathbb{N}) \\
\mathit{entryCondition}(\mathcal{V})(\mathit{base}, \mathit{end}, a, W) &= \{n \mid \forall n' < n. \forall W' \sqsupseteq W. \\
&\qquad (n', (\mathrm{rx}, \mathit{base}, \mathit{end}, a)) \in \mathcal{E}(\mathcal{V})(W')\}
\end{aligned}
$$

The following lemmas show that the above conditions are well-defined:

Lemma 26 (Read condition downwards-closed).

$$
\begin{aligned}
&\forall \mathcal{V}, n, n', W, \mathit{base}, \mathit{end}. \\
&\quad n \in \mathit{readCondition}(\mathcal{V})(\mathit{base}, \mathit{end}, W) \wedge \\
&\quad n' \leq n \\
&\qquad \Rightarrow n' \in \mathit{readCondition}(\mathcal{V})(\mathit{base}, \mathit{end}, W)
\end{aligned}
$$

HW

■

Lemma 27 (Read condition monotone in world).

$$
\begin{aligned}
&\forall \mathcal{V}, n, W, W', \mathit{base}, \mathit{end}. \\
&\quad (\mathit{base}, \mathit{end}, W') \sqsupseteq (\mathit{base}, \mathit{end}, W) \\
&\qquad \Rightarrow \mathit{readCondition}(\mathcal{V})(\mathit{base}, \mathit{end}, W') \supseteq \mathit{readCondition}(\mathcal{V})(\mathit{base}, \mathit{end}, W)
\end{aligned}
$$

■

Lemma 26

let $n' \leq n$ and assume

$$n \in readCondition(\mathcal{V})(b, e, W) \quad (\mathrm{I})$$

Show

$$n' \in readCondition(\mathcal{V})(b, e, W) \quad (\mathrm{II})$$

By (I) get $r, [b', e'] \supseteq [b, e]$ s.t.

$$W(r) \overset{n-1}{\subseteq} \iota_{b', e'}(\mathcal{V})$$

Use the same r, b', e' to show (II) i.e.

$$W(r) \overset{n'-1}{\subseteq} \iota_{b', e'}(\mathcal{V})$$

Result by downwards closure of $\overset{n}{\subseteq}$.

p. 22 1

Lemma

$A \overset{n}{\leq} B \wedge n' \leq n$

$\Rightarrow$

$A \overset{n'}{\leq} B$

Proof

Assume $A \overset{n}{\leq} B \wedge n' \leq n$

Show $A \overset{n'}{\leq} B$

$\Rightarrow (s_A, \phi_A) = (s_B, \phi_B)$

$\forall \hat{W} \in W.\ H_A\ s_A\ \hat{W} \overset{n}{=} H_B\ s_B\ \hat{W}$

given $\hat{W} \in W$ or show

$H_A\ s_A\ \hat{W} \overset{n'}{=} H_B\ s_B\ \hat{W}$

$\overset{n}{=}$ downwards closed.

Lemma 27

assume $(b', e', W') \sqsupseteq (b, e, W) \Rightarrow$ $\& \; (b,e) = (b'', e'')$

$W' \sqsupseteq W \quad (\mathrm{I})$

Show

$$rL(\mathcal{V})(\overset{b''}{b}, \overset{e''}{e}, W') \supseteq rL(\mathcal{V})(b, e, W)$$

let $n \in rL(\mathcal{V})(b, e, W)$

get $r, \ell \; [b', e'] \sqsupseteq [b, e]$ s.t.

$$W(r) \overset{n-1}{\subseteq} L_{b', e'} \quad (\mathrm{II})$$

use $r \; \ell \; [b', e']$ to show

$$n \in rL(\mathcal{V})(b, e, W')$$

need to show

$$W'(r) \overset{n-1}{\subseteq} L_{b', e'}(\mathcal{V})$$

By future worlds$^{(\mathrm{I})}$ we know

$(\phi', H') = (\phi, H)$ and $(s', s) \in \phi'$

where $W'(r) = (s', \phi', H)$ and $W(r) = (s, \phi, H)$

By (II) we know $s = (b', e')$ and $\phi = \text{'='}$ $H = H_{std}$

So s' must be (b', e')

Now we have $(s', \phi') = ((b', e'), =)$ it remains to be shown $\forall \hat{W} \in Wor$ $\underset{H_{std}}{\overset{\shortparallel}{H'}} \; \mathcal{V} \; \underset{(b', e')}{\overset{\shortparallel}{s'}} \; \hat{W} \; \hat{\subseteq} \; H_{std} \; \mathcal{V} \; (b, e) \; \hat{W}$

So the n-subset is satisfied. $\quad$ P:22 $\quad$ 2

Lemma 28 (Read condition non-expansive in worlds).

$$\forall \mathcal{V}, b, e, n, W, W'.$$
$$(b, e, W) \stackrel{n}{=} (b, e, W') \Rightarrow$$
$$\mathit{readCondition}(\mathcal{V})(b, e, W) \stackrel{n}{=} \mathit{readCondition}(\mathcal{V})(b, e, W')$$

– b', e'

HW

■

Lemma 29 (Read-write condition uniformity).

$$\forall \mathcal{V}, n, n', W, \mathit{base}, \mathit{end}.$$
$$n \in \mathit{readWriteCondition}(\mathcal{V})(\mathit{base}, \mathit{end}, W) \wedge$$
$$n' \leq n$$
$$\Rightarrow n' \in \mathit{readWriteCondition}(\mathcal{V})(\mathit{base}, \mathit{end}, W)$$

HW

■

Lemma 30 (Read-write condition monotone in world).

$$\forall \mathcal{V}, n, W, W', \mathit{base}, \mathit{end}.$$
$$(\mathit{base}, \mathit{end}, W') \sqsupseteq (\mathit{base}, \mathit{end}, W)$$
$$\Rightarrow \mathit{readWriteCondition}(\mathcal{V})(\mathit{base}, \mathit{end}, W') \supseteq \mathit{readWriteCondition}(\mathcal{V})(\mathit{base}, \mathit{end}, W)$$

– HW

RWC n.e. in Worlds missing.

HW

■

Lemma 31 (Execute condition downwards-closed).

$$\forall \mathcal{V}, n, n', W, \mathit{base}, \mathit{end}, \mathit{perm}.$$
$$n \in \mathit{executeCondition}(\mathcal{V})(\mathit{base}, \mathit{end}, \mathit{perm}, W) \wedge$$
$$n' \leq n$$
$$\Rightarrow n' \in \mathit{executeCondition}(\mathcal{V})(\mathit{base}, \mathit{end}, \mathit{perm}, W)$$

HW

■

Lemma 32 (Execute condition monotone in world).

$$\forall \mathcal{V}, n, W, W', \mathit{base}, \mathit{end}, \mathit{perm}.$$
$$(\mathit{base}, \mathit{end}, \mathit{perm}, W') \sqsupseteq (\mathit{base}, \mathit{end}, \mathit{perm}, W)$$
$$\Rightarrow \mathit{executeCondition}(\mathcal{V})(\mathit{base}, \mathit{end}, \mathit{perm}, W') \supseteq \mathit{executeCondition}(\mathcal{V})(\mathit{base}, \mathit{end}, \mathit{perm}, W)$$

HW

■

Lemma 33 (Execute condition non-expansive in worlds).

$$\forall \mathcal{V}, W_1, W_2, n, \mathit{base}, \mathit{end}, \mathit{perm}.$$
$$(\mathit{base}, \mathit{end}, \mathit{perm}, W_1) \stackrel{n}{=} (\mathit{base}, \mathit{end}, \mathit{perm}, W_2) \Rightarrow$$
$$\Rightarrow \mathit{executeCondition}(\mathcal{V})(b, e, p, W_1) \stackrel{n}{=} \mathit{executeCondition}(\mathcal{V})(b, e, p, W_2)$$

– HW

base and perm

■

Lemma 28 rL ne in Worlds

Assume $(b'', e'', W) \overset{n}{\simeq} (b, e, W) \Rightarrow$ $W \overset{n}{=} W'$ (I) and $(b,e) = (b'', e'')$

Show $rL(\mathcal{V})(b,e,W) \overset{n}{\simeq} rL(\mathcal{V})(b,e,W')$

let $k \leq n \in rL(\mathcal{V})(b,e,W)$

get r, $[b',e'] \supseteq [b,e]$ s.t.

$W(r) \overset{k-1}{\leq} L_{b',e'}\,\mathcal{V}$ (II)

Show

$W'(r) \overset{k-1}{\leq} L_{b',e'}\,\mathcal{V}$

From (I) have $W(r) \overset{n}{=} W'(r)$

$\Downarrow$

(III) $(s, \phi) = (s', \phi')$ and $H \overset{n}{=} H'$ (VI)

From (II) (IV) $s = (b', e')$ & $\phi = \text{"="}$ (V) &

$\forall \hat{W} \in Wor.\ H s \hat{W} \overset{k-1}{=} H_{std}\,\mathcal{V}\,(b',e')\,\hat{W}$

(III) & (IV) gives $s' = (b, e')$ (III) + (V) gives $\phi' = \text{"="}$

given $\hat{W} \in Wor$

From (VI) and (III) get $H\,s\,\hat{W} \overset{n}{=} H'\,s'\,\hat{W}$

$k-1 \leq n$ + downwards close $\overset{n}{=}$ gives

$H'\,s'\,\hat{W} \overset{k-1}{=} H_{std}\,\mathcal{V}(b',e')\,\hat{W}$

P. 23 1

Lemma 29 rW-cond uniform

Assume $n' \le n$ and $n \in rwC(\mathcal{V})(b,e,W)$ (I)

Show $n' \in rwC(\mathcal{V})(b,e,W)$

Get r, $[b',e'] \supseteq [b,e]$ from I_n s.t.:

$$W(r) \overset{n-1}{=} L_{b',e'} \mathcal{V} \quad \text{(II)}$$

Use r, and $[b',e']$, show

$$W(r) \overset{n'-1}{=} L_{b',e'} \mathcal{V}$$

From (II) get

$(s,\phi) \models ((b',e'),r) = |$ and $H \overset{n-1}{=} H_{std} \mathcal{V}$

Show

$(s,\phi) \models ((b',e'),r) = |$

$H \overset{n'-1}{=} H_{std} \mathcal{V}$ ✓

$n'-1 \le n-1$
$+$
downwards closure

p. 23 2

Lemma 30

rwC mono in worlds

assume $(b'', e'', W') \sqsupseteq (b, e, W) \Rightarrow \begin{cases} (b'', e'') \vdash (b, e) \\ W' \sqsupseteq W \quad (\mathrm{I}) \end{cases}$

Show

$$rwC(V)(\overset{b''}{b}, \overset{e''}{e}, W') \supseteq rwC(V)(b, e, W)$$

let $n \in rwC(V)(b, e, W)$ (II)

show $n \in rwC(V)(b, e, W')$

from (II) get r and $[b', e'] \supseteq [b, e]$ s.t.

$W(r) \overset{n-1}{=} \llcorner_{b', e'} V$ (III)

show

$W'(r) \overset{n-1}{=} \llcorner_{b', e'} V$

By (I) $H = H'$ $\Phi = \Phi'$ $(s, s') \in \Phi'$

By (III) $H \overset{n-1}{=} H_{std}$ ✓ $\Phi \equiv \doteq$ $s = (b', e')$

$\Downarrow$ $\Downarrow$

$H \overset{n-1}{=} H'$ $\Phi' \equiv \doteq$ $s' = (b', e')$

$\Downarrow$

$H' \overset{n-1}{=} H_{std}$ ✓

Lemma RWC n.e. in worlds:

$$\forall \mathcal{V}, b, e, n, W, W'. \quad (b,e,W) \overset{n}{\equiv} (b',e',W') \Rightarrow rwC(\mathcal{V})(b,e,W) \overset{n}{\equiv} rwC(\mathcal{V})(b',e',W)$$

Proof.

Assume $(b,e,W) \overset{n}{\equiv} (b'',e'',W') \Rightarrow (b,e) = (b'',e'')$ & $W \overset{n}{\equiv} W'$ (I)

let $k \in rwC(\mathcal{V})(b,e,W)$ for $k \leq n$

get r, $[b',e'] \supseteq [b,e]$ s.t.

$w(r) \overset{k-1}{\equiv} \iota_{b',e'}\mathcal{V}$ (II)

Show

$W'(r) \overset{k-1}{\equiv} \iota_{b',e'}\mathcal{V}$

From (I) $S = S'$, $\phi = \phi'$ and $H \overset{n}{\equiv} H'$

From (II) $S = (b',e')$, $\phi \equiv =$ and $H \overset{k-1}{\equiv} H_{std}\,\mathcal{V}$

$S' = (b',e')$ $\phi' \equiv =$ $\overset{=}{\Rightarrow}$ $H' \overset{k-1}{\equiv} H_{std}\,\mathcal{V}$

down-closed

P.23 4

Lemma 31

Exec Cond downwards-closed

let $n' \leq n$ and $n \in exec C(\mathcal{V})(b, e, p, W)$ (I)

show $n' \in exec C(\mathcal{V})(b, e, p, W)$

let $n'' < n'$, $W' \supseteq W$ and $a \in [b, e]$ be given.

show $(n'', (\cancel{\ }, b, e, a)) \in \mathcal{E}(\mathcal{V})(W')$

As $n' \leq n$ and $n'' < n'$ we have $n'' < n$. The result follows from assumption (I).

p. 28

5

Lemma 32 execL mono in worlds.

Assume $(b', e', p', W_2) \geq (b, e, p, W_1) \Rightarrow \& \begin{array}{l}(b,e,p) = (b',e',p') \\ W_2 \sqsupseteq W_1 \quad (\mathrm{II})\end{array}$

Show

$$\mathrm{execL}(\mathcal{V})(\underset{b}{\overset{\shortparallel}{b'}}, \underset{e}{\overset{\shortparallel}{e'}}, \underset{p}{\overset{\shortparallel}{p'}}, W_2) \geq \mathrm{execL}(\mathcal{V})(b, e, p, W_1)$$

let $n \in \mathrm{execL}(\mathcal{V})(b, e, p, W_1)$ (I)

let $n' < n$, $W_2' \sqsupseteq W_2$ and $a \in [b, e]$ be given

By trans and (II) $W_2' \sqsupseteq W_1$. Result follows by using (I).

p. 23 6

Lemma 33

let $(b', e', p', W_1) \overset{n}{\simeq} (b, e, p, W_2) \Rightarrow (b', e', p') = (b, e, p)$

& $W_1 \overset{n}{\simeq} W_2$ (I)

let $n \vDash exec(\mathcal{V})(b, e, p, W_1)$ (III)

show $n \vDash exec(\mathcal{V})(b, e, p, W_2)$

let $n' < n$, $W_2' \sqsupseteq W_2$ (II), $a \in [b, e]$ be given.

By lemma 45 due to (I) and (II) get

W_1' s.t. $W_1' \sqsupseteq W_1$ (IV) . $W_1' \overset{n}{\simeq} W_2'$

From assumption (III) using (IV), n', and a get

$(n', (p, b, e, a)) \in \mathcal{E}(\mathcal{V})(W_1')$

Since $\mathcal{E}(\mathcal{V})$ n.e. we have

$\mathcal{E}(\mathcal{V})(W_1') \overset{n}{\simeq} \mathcal{E}(\mathcal{V})(W_2')$

and further as $n' < n$ and

$(n', (p, b, e, a)) \in \mathcal{E}(\mathcal{V})(W_2')$.

p. 23 7

Lemma 34 (Entry condition downwards-closed).

$$\begin{aligned}&\forall \mathcal{V}, n, n', W, base, end, a.\\ &\quad n \in entryCondition(\mathcal{V})(base, end, a, W) \wedge\\ &\quad n' \leq n\\ &\qquad \Rightarrow n' \in entryCondition(\mathcal{V})(base, end, a, W)\end{aligned}$$

like lemma 31

■

Lemma 35 (Entry condition monotone in world).

$$\begin{aligned}&\forall \mathcal{V}, n, W, W', base, end, a.\\ &\quad (base, end, a, W') \sqsupseteq (base, end, a, W)\\ &\qquad \Rightarrow entryCondition(\mathcal{V})(base, end, perm, W') \supseteq entryCondition(\mathcal{V})(base, end, perm, W)\end{aligned}$$

like lemma 32

■

Lemma 36 (Entry condition non-expansive in worlds).

$$\begin{aligned}&\forall \mathcal{V}, W_1, W_2, n, base, end, a.\\ &\quad (base, end, perm, W_1) \stackrel{n}{=} (base, end, perm, W_2) \Rightarrow\\ &\qquad \Rightarrow executeCondition(\mathcal{V})(b, e, p, W_1) \stackrel{n}{=} executeCondition(\mathcal{V})(b, e, p, W_2)\end{aligned}$$

like lemma 33

■

Finally, we need to show that all the conditions are non-expansive, but we later want to use Banach's fixed point theorem to define the value relation. For this we will need that the above conditions are contractive, and if they are contractive, then they are also non-expansive, so we show that each of the conditions are contractive:

Lemma 37 (Read condition contractive).

$$\begin{aligned}&\forall \mathcal{V}, \mathcal{V}', n.\\ &\quad \mathcal{V} \stackrel{n}{=} \mathcal{V}' \Rightarrow readCondition(\mathcal{V}) \stackrel{n+1}{=} readCondition(\mathcal{V}')\end{aligned}$$

HW

■

Lemma 38 (Write condition contractive).

$$\begin{aligned}&\forall \mathcal{V}, \mathcal{V}', n.\\ &\quad \mathcal{V} \stackrel{n}{=} \mathcal{V}' \Rightarrow readWriteCondition(\mathcal{V}) \stackrel{n+1}{=} readWriteCondition(\mathcal{V}')\end{aligned}$$

HW

■

Lemma 39 (Execute condition contractive).

$$\begin{aligned}&\forall \mathcal{V}, \mathcal{V}', n.\\ &\quad \mathcal{V} \stackrel{n}{=} \mathcal{V}' \Rightarrow executeCondition(\mathcal{V}) \stackrel{n+1}{=} executeCondition(\mathcal{V}')\end{aligned}$$

HW

■

Lemma 37

rC contractive in $\mathcal{V}$

assume $\mathcal{V} \overset{n}{\equiv} \mathcal{V}'$

show $rC(\mathcal{V}) \overset{n+1}{\equiv} rC(\mathcal{V}')$

To this end let (b,e,W) be given and show

$$rC(\mathcal{V})(b,e,W) \overset{n+1}{\equiv} rC(\mathcal{V})(b,e,W)$$

let $k \in rC(\mathcal{V})(b,e,W)$ for $k < n+1$

and show $k \in rC(\mathcal{V}')(b,e,W)$

get $r, (b',e'] \supseteq [b,e]$ s.t.

$$W(r) \overset{k-1}{\subseteq} \iota_{b',e'} \mathcal{V} \qquad (I)$$

Show (using r, b', e')

$$W(r) \overset{k-1}{\subseteq} \iota_{b',e'} \mathcal{V}'$$

From (I), we have $s = (b',e')$, $\phi \equiv =$, and $\forall \hat{W} \in Wor_{\mathcal{V}}$. $H\, s\, \hat{W} \overset{k-1}{\subseteq} H_{std}(b',e')$

(lemma)

H_{std} is n.e. in val. rel., so

$$H_{std}\, \mathcal{V} \overset{n}{\equiv} H_{std}\, \mathcal{V}'$$

As $k < n+1$, we have $k-1 < n$. By downwards closure of $\overset{n}{\equiv}$, we get

$$H_{std}\, \mathcal{V} \overset{k-1}{\equiv} H_{std}\, \mathcal{V}'$$

Given $\hat{W}$, we have

$$H\, s\, \hat{W} \overset{k-1}{\subseteq} H_{std}\, \mathcal{V}\, (b',e')\, \hat{W} \overset{k-1}{\equiv} H_{std}\, \mathcal{V}'\, (b',e')\, \hat{W}$$

Lemma 38 rwCond contractive in $\mathcal{V}$.

Assume $\mathcal{V} \overset{n}{=} \mathcal{V}'$

show $rwCond(\mathcal{V}) \overset{n+1}{=} rwCond(\mathcal{V}')$

let b, e, W be given and let $k < n+1$ s.t.

$$k \in rwCond(\mathcal{V})(b, e, W) \quad (I)$$

Show

$$k \in rwCond(\mathcal{V}')(b, e, W)$$

From (I), get $r, [b', e'] \supseteq [\![b, e]\!]$ s.t.

$$W(r) \overset{k-1}{=} \iota_{b', e'} \mathcal{V} \quad (II)$$

Using r and b', e', we need to show

$$W(r) \overset{k-1}{=} \iota_{b', e'} \mathcal{V}'$$

As $\iota_{b', e'}$ is n.c in $\mathcal{V}$ (lemma), we have

$$\iota_{b', e'} \mathcal{V} \overset{n}{=} \iota_{b', e'} \mathcal{V}'$$

As $k < n+1$ we have $k-1 < n$ and by downwards closure of $\overset{n}{=}$, we get

$$\iota_{b', e'} \mathcal{V} \overset{k-1}{=} \iota_{b', e'} \mathcal{V}'$$

So

$$W(r) \overset{k-1}{\underset{(II)}{=}} \iota_{b', e'} \mathcal{V} \overset{k-1}{=} \iota_{b', e'} \mathcal{V}'.$$

Lemma 34 Execute cond. contractive in $\mathcal{V}$

Assume $\mathcal{V} \overset{n}{=} \mathcal{V}'$

Show $execCond(\mathcal{V}) \overset{n+1}{=} execCond(\mathcal{V}')$

let b, e, p, W be given and take $k < n+1$ s.t.

$$k \in execCond(\mathcal{V})(b, e, p, W) \quad (\mathrm{I})$$

show

$$k \in execCond(\mathcal{V}')(b, e, p, W)$$

To this end, let

$k' < k$, $W' \sqsupseteq W$, and $a \in [b, e]$ be given.

and show

$$(k', (p, b, e, a)) \in \mathcal{E}(\mathcal{V}')(W)$$

using (I) w/ k', W', and a, get

$$(k', (p, b, e, a)) \in \mathcal{E}(\mathcal{V})(W) \quad (\mathrm{II})$$

As $\mathcal{E}$ is n.e in $\mathcal{V}$, we get

$$\mathcal{E}(\mathcal{V}) \overset{n}{=} \mathcal{E}(\mathcal{V}') \quad (\mathrm{III})$$

which in turn gives

$$\mathcal{E}(\mathcal{V})(W) \overset{n}{=} \mathcal{E}(\mathcal{V}')(W)$$

As $k' < k < n+1$ we must have $k' < n$, so (III) and (II) give us

$$(k', (p, b, e, a)) \in \mathcal{E}(\mathcal{V}')(W)$$

p. 24 3

Lemma 40 (Entry condition contractive).

$$\forall \mathcal{V}, \mathcal{V}', n.$$
$$\mathcal{V} \stackrel{n}{=} \mathcal{V}' \Rightarrow \mathit{entryCondition}(\mathcal{V}) \stackrel{n+1}{=} \mathit{entryCondition}(\mathcal{V}')$$

■

Like lemma 39.

2.3.7 Value Relation

The value relation, is defined as follows:

$$\mathcal{V} : (\text{World} \xrightarrow{\text{mon, ne}} \text{UPred}(\mathit{Words})) \xrightarrow{\text{ne}} \text{World} \xrightarrow{\text{mon, ne}} \text{UPred}(\text{Word})$$

$$\begin{aligned}
\mathcal{V} \stackrel{\text{def}}{=} \lambda\, V. \lambda\, W.\ & \{(n, i) \mid i \in \mathbb{Z}\} \cup \\
& \{(n, (\text{o}, \mathit{base}, \mathit{end}, a))\} \cup \\
& \{(n, (\text{ro}, \mathit{base}, \mathit{end}, a)) \mid n \in \mathit{readCondition}(V)(\mathit{base}, \mathit{end}, W)\} \cup \\
& \{(n, (\text{rw}, \mathit{base}, \mathit{end}, a)) \mid n \in \mathit{readWriteCondition}(V)(\mathit{base}, \mathit{end}, W)\} \cup \\
& \{(n, (\text{rx}, \mathit{base}, \mathit{end}, a)) \mid \\
& \quad n \in \mathit{readCondition}(V)(\mathit{base}, \mathit{end}, W) \wedge \\
& \quad n \in \mathit{executeCondition}(V)(\mathit{base}, \mathit{end}, \text{rx}, W)\} \cup \\
& \{(n, (\text{e}, \mathit{base}, \mathit{end}, a)) \mid n \in \mathit{entryCondition}(V)(\mathit{base}, \mathit{end}, a, W)\} \cup \\
& \{(n, (\text{rwx}, \mathit{base}, \mathit{end}, a)) \mid \\
& \quad n \in \mathit{readWriteCondition}(V)(\mathit{base}, \mathit{end}, W) \wedge \\
& \quad n \in \mathit{executeCondition}(V)(\mathit{base}, \mathit{end}, \text{rx}, W) \wedge \\
& \quad n \in \mathit{executeCondition}(V)(\mathit{base}, \mathit{end}, \text{rwx}, W)\}
\end{aligned}$$

Lemmas missing contractive, + DC ne, mon, ne. The missing lemmas written on the following pages.

2.4 Standard Regions

To define the value relation, we use a standard heap invariant that ensures all values in the region are in the value relation. The following region uses pairs of natural numbers, $(\mathit{base}, \mathit{end})$, as states, so pairs of natural numbers are in the set State.

$$\iota_{\mathit{start},\mathit{end}} : \text{Region}$$

$$\iota_{\mathit{base},\mathit{end}} \stackrel{\text{def}}{=} ((\mathit{base}, \mathit{end}), =, H_{\mathit{std}})$$

$$H_{\mathit{std}}\ (\mathit{base}, \mathit{end})\ W \stackrel{\text{def}}{=} \left\{ (n, hs) \,\middle|\, \begin{array}{l} \mathrm{dom}(hs) = [\mathit{base}, \mathit{end}] \wedge \\ \forall a \in [\mathit{base}, \mathit{end}].\, (n-1, hs(a)) \in \mathcal{V}(\xi\, W) \end{array} \right\}$$

Note that this region is defined in terms of the value relation, and the value relation is defined in terms of this invariant. We define the well-definedness lemma here, but show it in the appendix.

Lemma 41 ($\iota_{\mathit{start},\mathit{end}}$ is well-defined). *For all base and end, $H_{\mathit{std}}\ (\mathit{base}, \mathit{end})$ is monotone and non-expansive. $=$ is a reflexive and transitive relation.* ■

Lemma a) $\mathcal{V}$ non-expansive in V

for all V, V',

$$V \overset{n}{=} V'$$
$$\Longrightarrow$$
$$\mathcal{V}\, V \overset{n}{=} \mathcal{V}\, V'$$

Lemma b) $\mathcal{V}$ non-expansive in W

for all V, W, W'

$$W \overset{n}{=} W'$$
$$\Longrightarrow$$
$$\mathcal{V}\, V\, W \overset{n}{=} \mathcal{V}\, V\, W$$

Lemma c) $\mathcal{V}$ mono in W.

for all V, W, W'

$$W' \supseteq W$$
$$\Longrightarrow$$
$$\mathcal{V}\, V\, W' \supseteq \mathcal{V}\, V\, W$$

Lemma d) $\mathcal{V}$ contractive in V.

for all V, V'

$$V \overset{n}{=} V'$$
$$\Longrightarrow$$
$$\mathcal{V}\, V \overset{n+1}{=} \mathcal{V}\, V$$

Lemma e) $\mathcal{V}$ downwardsclosed

for all V, W, n, n', c

$n' \leq n$ and $(n, c) \in \mathcal{V}\, V\, W$

$\Longrightarrow (n', c) \in \mathcal{V}\, V\, W$

Lemma a) proof

Assume $V \stackrel{n}{=} V'$ and let W be given.

Show

$$\mathcal{V}\, V\, W \stackrel{n}{=} \mathcal{V}\, V'\, W.$$

To this end let

$(k,c) \in \mathcal{V}\, V\, W$ for $k<n$

and show

$(k,c)\ \mathcal{V}\, V'\, W.$

Continue by case on c.

For $c = (ro, b, e, a)$

we know $k \in r\mathcal{C}(V)(b,e,W)$ (I)

and need to show $k \in r\mathcal{C}(V')(b,e,W)$.

This follows by $r\mathcal{C}$ being ne in V, so

$$r\mathcal{C}(V)(b,e,W) \stackrel{n}{=} r\mathcal{C}(V')(b,e,W)$$

and as $k<n$ and (I) we may conclude

$k \in r\mathcal{C}(V')(b,e,W)$.

The remaining cases follows from readCondition, readWriteCondition, executeCondition, and entry condition all being non-expansive in the value relation.

Lemma b) $\mathcal{V}$ non-expansive in W.

Assume $W \overset{n}{=} W'$ (I)

Show $\mathcal{V}\, V\, W \overset{n}{=} \mathcal{V}\, V\, W'$

to this end let $k<n$ and c be given s.t.

$(k,c) \in \mathcal{V}\, V\, W$

Show
$(k,c) \in \mathcal{V}\, V\, W'$

proceed by cases on c.

If $c = (ro, b, e, a)$, then we know

$k \in rC(V)(b,e,W)$ (II)

and need to show

$k \in rC(V)(b,e,W')$ (III)

From (I), we can get $(b,e,W) \overset{n}{=} (b,e,W')$ which allows us to use that $rC(V)$ is non-expansive. to conclude:

$$rC(V)(b,e,W) \overset{n}{=} rC(V)(b,e,W')$$

as $k<n$ and we have (II), we get (III).

The remaining cases either follows trivially or by readCondition, readWriteCondition, execCondition, and entryCondition being non-expansive (all lemmas)

p. 25 3

Lemma c) $\mathcal{V}$ mono in W.

Assume $W' \sqsupseteq W$ (II)

Show $\mathcal{V}\, V\, W' \supseteq \mathcal{V}\, V\, W$

To this end let (n,c) be given s.t.

$(n,c) \in \mathcal{V}\, V\, W$ (I)

and show

$(n,c) \in \mathcal{V}\, V\, W'$

proceed by cases. If $c = (r_0, b, e, a)$, then (I) gives

$n \in \text{readCondition}(V)(b,e,W)$ and (II) gives us $(b,e,W') \sqsupseteq (b,e,W)$

By monotonicity, we know that

$\text{readCondition}(V)(b,e,W') \supseteq \text{readCondition}(V)(b,e,W)$

So $n \in \text{readCondition}(V)(b,e,W')$ is true.

The remaining cases are either trivial or follows trivially from monotonicity of readCondition, readWriteCondition, executeCondition, and entryCondition.

p.25 4

Lemma d) $\mathcal{V}$ contractive in V

Assume $V \overset{n}{=} V'$

Show $\mathcal{V}\,V \overset{n+1}{=} \mathcal{V}\,V'$

To this end let $k < n+1$, W, and c be given s.t

$(k,c) \in \mathcal{V}\,V\,W$ (*)

show

$(k,c) \in \mathcal{V}\,V'\,W$

Proceed by case on c. If $c = (ro, b, e, a)$, then by (*) we know

$k \in readCondition(V)(b,e,W)$

As readCondition is contractive in V, and $V \overset{n+1}{=} \ldots$ we know

$readCondition(V)(b,e,W) \overset{n+1}{=} readCondition(V')(b,e,W)$

And as $k < n+1$ we may conclude that

$(k,c) \in readCondition(V')(b,e,W)$

The remaining cases are either trivial or follows from readCondition, readWriteCondition, execCondition, and entryCondition being contractive in V.

Lemma e) $\mathcal{V}$ downwards closed

Assume $n' \leq n$ (7) and $(n,c) \in \mathcal{V}\ V\ W$

Show $(n',c) \in \mathcal{V}\ V\ W$.

Proceed by case on c.

If $c = (ro, b, e, a)$, then

SPTS. $n' \in$ readCondition$(V)(b,e,W)$

By assumption we have

$n \in$ readCondition$(V)(b,e,W)$

using this and $n' \leq n$, we get the desired result from downwards closure of readCondition.

The remaining cases are trivial or follows from readCondition, readWriteCondition, executeCondition, and entryCondition being downwardsclosed.

Lemma 48 (Write condition implies read condition).

$$\forall n, W, base, end.$$
$$readWriteCondition(n, W, base, end) \Rightarrow readCondition(n, W, base, end)$$

■

[Proof of lemma 48]

Proof. Follows directly from the definition. □

Lemma 49 (Value relation uniformity).

$$\forall n' < n. \forall W. \forall w.$$
$$(n, w) \in \mathcal{V}(W) \Rightarrow (n', w) \in \mathcal{V}(W)$$

■

Proof. Follows from the uniformity of *readCondition*, *readWriteCondition*, *executeCondition*, and *entryCondition*. □

Lemma 50 (Value relation monotone in worlds).

$$\forall n. \forall W' \sqsupseteq W. \forall w.$$
$$(n, w) \in \mathcal{V}(W) \Rightarrow (n, w) \in \mathcal{V}(W')$$

■

Proof. Follows from uniformity of *readCondition*, *readWriteCondition*, *executeCondition*, and *entryCondition* in the worlds. That is Lemma 27, 30, 32, and 35. □

Lemma 51 (Value relation contractive).

$$\forall n. \forall W \in \text{World} \forall w.$$
$$(n, w) \in \mathcal{V}(W) \Rightarrow$$
$$(n + 1, w) \in \mathcal{V}(W)$$

■

Proof. □

Proof of lemma 11. Let $n' < n$, W, reg, and hs be given. Assume $(n, (reg, hs)) \in \mathcal{O}(W)$. Let $heap_f$, $heap'$, $i \leq n'$ be given and assume $(reg, hs \uplus heap_f) \to_i (halted, heap')$. By assumption, we have a $W' \sqsupseteq W$ and hs' such that

$$heap' = hs' \uplus heap_f \tag{11}$$
$$hs' :_{n-i} W' \tag{12}$$

. Using W' and hs' as existential witnesses, we already have Equation 11 as the first necessary condition and from the above heap satisfaction along with heap satisfaction being uniform in n, we get $hs' :_{n'-i} W'$. These are the two conditions necessary to get $(n', (reg, hs)) \in \mathcal{O}(W)$. □