노트 제목 2010-07-09

# A Kripke Logical relation between ML & Assembly


Chung-Kil Hur & Derek Dreyer

PPS MPI-SWS


Sep 2010

@ Dagstuhl

# observational equivalence between two languages

$$\mathcal{L}_1 \qquad\qquad \mathcal{L}_2$$

$$P_1 \quad \approx \quad P_2$$

Q: A good notion of program equivalence between $\mathcal{L}_1$ and $\mathcal{L}_2$?

Requirements

① type-indexed relations e.g.) $\lambda x.x \approx_{int \to int} \lambda x.x+0$

② adequate : $P_1 \approx_T P_2 \Rightarrow obs(load_{\mathcal{L}_1}(P_1)) = obs(load_{\mathcal{L}_2}(P_2)) \neq error$

③ compositional : $P_1 \approx_T P_2 \wedge K_1 \approx_{T \to S} K_2 \Rightarrow link_{\mathcal{L}_1}(K_1, P_1) \approx_S link_{\mathcal{L}_2}(K_2, P_2)$

④ extensional (informal concepts) : sufficiently populated

# Application: Compositional Compiler Correctness

$$
\begin{array}{lcccc}
 & \forall p. & \forall p. & \\
\mathcal{H} \text{ (e.g. ML)} & p : T & p : T & p \\
\approx_T & \approx_T & \approx_T & \approx_T \\
\mathcal{L} \text{ (e.g. assembly)} & \llbracket p \rrbracket_1 & \llbracket p \rrbracket_2 & m & \leftarrow \text{hand-optimized code}
\end{array}
$$

$$
\begin{array}{c} p \\ \approx_T \\ \llbracket p \rrbracket \end{array} \wedge \begin{array}{c} q \\ \approx_{T \to S} \\ m \end{array}
\xrightarrow{\text{by comp}}
\begin{array}{c} \mathrm{link}_{\mathcal{H}}(p, q) \\ \approx_S \\ \mathrm{link}_{\mathcal{L}}(\llbracket p \rrbracket, m) \end{array}
\xrightarrow{\text{by adq}}
\begin{array}{c} \mathrm{obs}(\mathrm{load}_{\mathcal{H}}(\mathrm{link}_{\mathcal{H}}(p, q))) \\ = \\ \mathrm{obs}(\mathrm{load}_{\mathcal{L}}(\mathrm{link}_{\mathcal{L}}(\llbracket p \rrbracket, m))) \\ \neq \text{error} \end{array}
$$

# Overview

Language-generic logical relation

Logical relation between Assembly and $\lambda^{\forall,\exists,\mu,\text{ref}}$

- self modifying code
- garbage collector (Mark-Sweep, Copying)

Adq & Comp ⇝ Compos. Compiler Correct.

Based on

- Logical relation between SECD and $\lambda^{\text{fix},\forall,\exists}$

  (ICFP'09, MSR techrep: Benton & Hur) ⇝ Basic idea of Compositional Compiler Correctness

- Logical relation on $\lambda^{\forall,\exists,\mu,\text{ref}}$

  (ICFP'10: Dreyer, Neis & Birkedal) ⇝ possible worlds model as STS with priv. vs pub.

  - step-indexing (Appel & McAllester)
  - biorthogonality (Krivine; Pitts & Stark)

## Language : High

**HIGH** – Syntax & Semantics

$$
\begin{array}{lll}
\tau & ::= & \alpha \mid b \mid \tau_1 \times \tau_2 \mid \tau_1 \to \tau_2 \mid \forall\alpha.\,\tau \mid \exists\alpha.\,\tau \mid \mu\alpha.\,\tau \mid \mathsf{ref}\ \tau \\
e & ::= & x \mid \ell \mid \langle e_1, e_2\rangle \mid e.1 \mid e.2 \mid \lambda x{:}\tau.\,e \mid e_1\ e_2 \mid \Lambda\alpha.e \mid e\ \tau \mid \\
 & & \mathsf{pack}\ \langle \tau_1, e\rangle\ \mathsf{as}\ \tau_2 \mid \mathsf{unpack}\ e_1\ \mathsf{as}\ \langle\alpha, x\rangle\ \mathsf{in}\ e_2 \mid \\
 & & \mathsf{roll}_\tau\ e \mid \mathsf{unroll}\ e \mid \mathsf{ref}\ e \mid e_1 := e_2 \mid {!e} \mid e_1 == e_2 \mid \ldots \\
v & ::= & x \mid \ell \mid \langle v_1, v_2\rangle \mid \lambda x{:}\tau.\,e \mid \Lambda\alpha.e \mid \mathsf{pack}\ \langle\tau_1, v\rangle\ \mathsf{as}\ \tau_2 \mid \mathsf{roll}_\tau\ v \mid \ldots \\
K & ::= & \bullet \mid \langle K, e_2\rangle \mid \langle v_1, K\rangle \mid K.1 \mid K.2 \mid K\ e_2 \mid v_1\ K \mid K\ \tau \mid \mathsf{roll}_\tau\ K \mid \\
 & & \mathsf{unroll}\ K \mid \mathsf{pack}\ \langle\tau_1, K\rangle\ \mathsf{as}\ \tau_2 \mid \mathsf{unpack}\ K\ \mathsf{as}\ \langle\alpha, x\rangle\ \mathsf{in}\ e_2 \mid \\
 & & \mathsf{ref}\ K \mid K := e_2 \mid v_1 := K \mid {!K} \mid K == e_2 \mid v_1 == K \mid \ldots
\end{array}
$$

$$\Sigma ::= \cdot \mid \Sigma, \ell{:}\tau \text{ with } \mathrm{ftv}(\tau) = \emptyset \qquad \Delta ::= \cdot \mid \Delta, \alpha \qquad \Gamma ::= \cdot \mid \Gamma, x{:}\tau$$

Static semantics : $\Sigma; \Delta; \Gamma \vdash e : \tau$

$$\mathrm{HCVal} \stackrel{\text{def}}{=} \{\, v \mid \mathrm{ftv}(v) = \emptyset \wedge \mathrm{fv}(v) = \emptyset \,\}$$

$$\mathrm{HHeap} \stackrel{\text{def}}{=} \{\, h \in \mathrm{HLoc} \rightharpoonup_{\mathrm{fin}} \mathrm{HCVal} \,\}$$

Dynamic semantics : $(h, e) \hookrightarrow (h', e')$

# Language : Low

**LOW** – Syntax

$$\mathrm{PConf} \stackrel{\mathrm{def}}{=} \{\, (\Phi, \mathrm{pc}) \in \mathrm{PMem} \times \mathrm{PAddr} \,\}$$

$$\begin{aligned}\mathrm{PMem} \stackrel{\mathrm{def}}{=} \{\, \Phi = (\mathrm{code}, \mathrm{reg}, \mathrm{stk}, \mathrm{hp}) \\ \in \mathrm{PCode} \times \mathrm{RegFiles} \times \mathrm{Stack} \times \mathrm{Heap} \,\}\end{aligned}$$

$$\mathrm{PCode} \stackrel{\mathrm{def}}{=} \mathrm{PAddr} \to \mathrm{Instruction} \qquad \mathrm{PRegFile} \stackrel{\mathrm{def}}{=} \mathrm{Register} \to \mathrm{PWord}$$

$$\mathrm{PStack} \stackrel{\mathrm{def}}{=} \mathrm{PAddr} \to \mathrm{PWord} \qquad \mathrm{PHeap} \stackrel{\mathrm{def}}{=} \mathrm{PAddr} \to \mathrm{PWord}$$

$$\mathrm{PAddr} \stackrel{\mathrm{def}}{=} \{\, a \in \mathbb{N} \,\} \qquad \mathrm{PWord} \stackrel{\mathrm{def}}{=} \{\, w \in \{\, 0, 1 \,\} \times \mathbb{N} \,\}$$

$$r \in \mathrm{Register} ::= \mathrm{sp} \mid \mathrm{sv}_0 \mid \ldots \mid \mathrm{sv}_4 \mid \mathrm{wk}_0 \mid \ldots \mid \mathrm{wk}_5$$

$$\mathrm{lv} \in \mathrm{PLvalue} ::= \lfloor r \rfloor \mid \langle a \rangle_{\mathrm{s}} \mid \langle r - o \rangle_{\mathrm{s}} \mid \langle a \rangle_{\mathrm{h}} \mid \langle r + o \rangle_{\mathrm{h}}$$

$$\mathrm{rv} \in \mathrm{PRvalue} ::= \mathrm{lv} \mid w$$

$$\begin{aligned}\iota \in \mathrm{Instruction} ::= {} & \texttt{fail} \mid \texttt{halt} \mid \texttt{jmp}\ \mathrm{rv} \mid \texttt{jnz}\ \mathrm{rv}\ \mathrm{rv} \mid \texttt{jneq}\ \mathrm{rv}\ \mathrm{rv}\ \mathrm{rv} \mid \\ & \texttt{jptr}\ \mathrm{rv}\ \mathrm{rv} \mid \texttt{setptr}\ \mathrm{lv} \mid \texttt{move}\ \mathrm{lv}\ \mathrm{rv} \mid \texttt{plus}\ \mathrm{lv}\ \mathrm{rv}\ \mathrm{rv} \mid \\ & \texttt{minus}\ \mathrm{lv}\ \mathrm{rv}\ \mathrm{rv} \mid \texttt{isr}\ \mathrm{lv}\ \mathrm{rv} \mid \texttt{isw}\ \mathrm{rv}\ \mathrm{rv}\end{aligned}$$

# Awkward example

$$\text{let } x = \text{ref } 0 \text{ in } \lambda f : \text{unit} \to \text{unit}.\ (x := 1\ ;\ f\,\langle\rangle\ ;\ !x)$$

$$\approx$$

$$\lambda f : \text{unit} \to \text{unit}.\ (f\,\langle\rangle\ ;\ 1)$$

Motivating example

$p \stackrel{\text{def}}{=} \lambda\,\text{alloc}, \text{bg}.$ [

| | | | | | |
|---|---|---|---|---|---|
| $\text{bg}$ | `move` | $\lfloor \text{wk}_4 \rfloor$ | $\underline{\text{bg}+3}$ | | |
| | `move` | $\lfloor \text{wk}_5 \rfloor$ | $\underline{1}$ | | |
| | `jmp` | $\underline{\text{alloc}}$ | | | |
| $\text{bg}+3$ | `move` | $\langle \text{wk}_5 + 0 \rangle_{\text{h}}$ | $\underline{\text{bg}+5}$ | | |
| | `jmp` | $\lfloor \text{wk}_0 \rfloor$ | | | |

create a closure and return it

| | | | | | |
|---|---|---|---|---|---|
| $\text{bg}+5$ | ~~`move`~~ | $\cancel{\lfloor \text{wk}_3 \rfloor}$ | $\cancel{\underline{\text{bg}+10}}$ | jmp $\underline{\text{bg}+12}$ | |
| $\text{bg}+6$ | `isr` | $\lfloor \text{wk}_4 \rfloor$ | $\lfloor \text{wk}_3 \rfloor$ | | |
| | `minus` | $\lfloor \text{wk}_4 \rfloor$ | $\lfloor \text{wk}_4 \rfloor$ | $\underline{666}$ | |
| | `isw` | $\lfloor \text{wk}_3 \rfloor$ | $\lfloor \text{wk}_4 \rfloor$ | | |
| | `plus` | $\lfloor \text{wk}_3 \rfloor$ | $\lfloor \text{wk}_3 \rfloor$ | $\underline{1}$ | |
| $\text{bg}+10$ | $\mathbb{D}(\mathbb{E}($`jneq` | $\text{bg}+6$ | $\lfloor \text{wk}_3 \rfloor$ | $\underline{\text{bg}+21}$ | $)+666)$ |
| $\text{bg}+11$ | $\mathbb{D}(\mathbb{E}($`isw` | $\underline{\text{bg}+5}$ | $\underline{\mathbb{E}(\texttt{jmp}\ \text{bg}+12)}$ | | $)+666)$ |
| $\text{bg}+12$ | $\mathbb{D}(\mathbb{E}($`move` | $\langle \text{wk}_1 + 0 \rangle_{\text{h}}$ | $\underline{\text{bg}+13}$ | | $)+666)$ |

decoding

| | | | | | |
|---|---|---|---|---|---|
| $\text{bg}+13$ | $\mathbb{D}(\mathbb{E}($`plus` | $\lfloor \text{sp} \rfloor$ | $\lfloor \text{sp} \rfloor$ | $\underline{1}$ | $)+666)$ |
| | $\mathbb{D}(\mathbb{E}($`move` | $\langle \text{sp} - 1 \rangle_{\text{s}}$ | $\lfloor \text{wk}_0 \rfloor$ | | $)+666)$ |
| | $\mathbb{D}(\mathbb{E}($`move` | $\lfloor \text{wk}_1 \rfloor$ | $\lfloor \text{wk}_2 \rfloor$ | | $)+666)$ |
| | $\mathbb{D}(\mathbb{E}($`move` | $\lfloor \text{wk}_0 \rfloor$ | $\underline{\text{bg}+18}$ | | $)+666)$ |
| | $\mathbb{D}(\mathbb{E}($`jmp` | $\langle \text{wk}_1 + 0 \rangle_{\text{h}}$ | | | $)+666)$ |
| $\text{bg}+18$ | $\mathbb{D}(\mathbb{E}($`move` | $\lfloor \text{wk}_5 \rfloor$ | $\underline{1}$ | | $)+666)$ |
| | $\mathbb{D}(\mathbb{E}($`minus` | $\lfloor \text{sp} \rfloor$ | $\lfloor \text{sp} \rfloor$ | $\underline{1}$ | $)+666)$ |
| $\text{bg}+20$ | $\mathbb{D}(\mathbb{E}($`jmp` | $\langle \text{sp} - 0 \rangle_{\text{s}}$ | | | $)+666)$ ] |

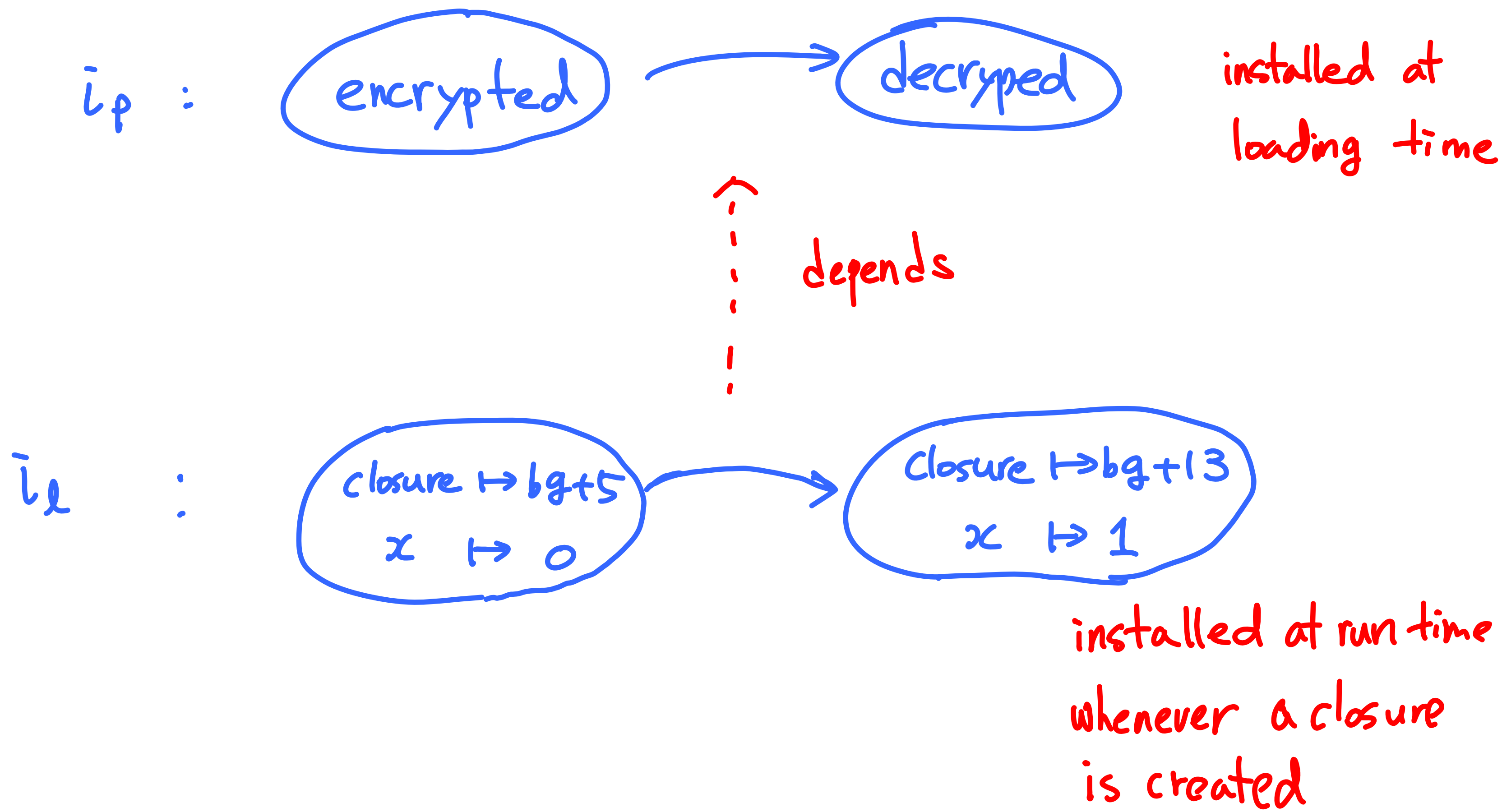
$\lambda f.\ f\langle\rangle; 1$

## Key ideas

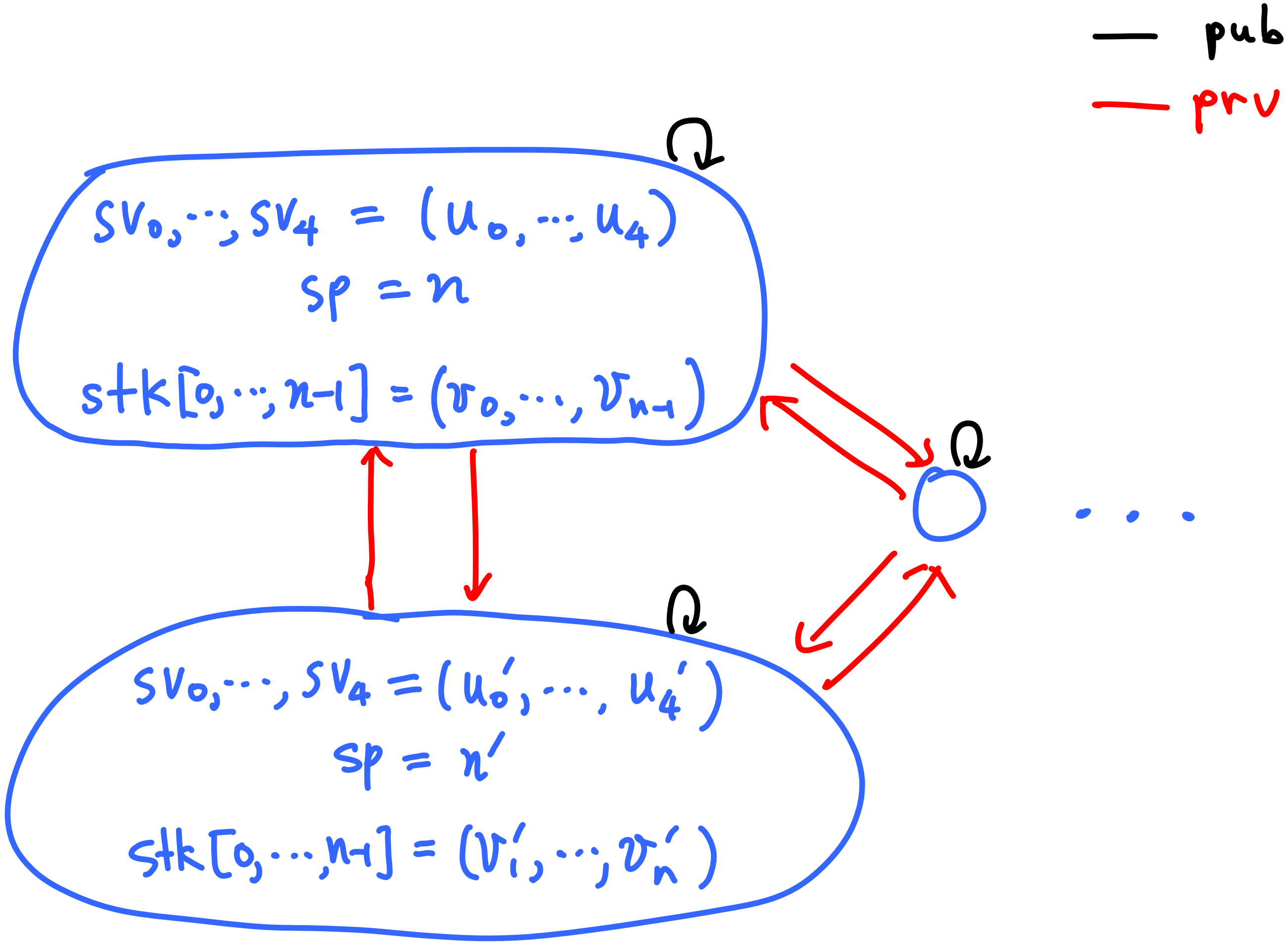
- Island for program code
- Island for closures and local states
- Island for stack and registers
- Garbage Collection

# Islands for code, closures & local states

$e = \text{let } x := \text{ref } 0 \text{ in } (x := 1;\ f\,\langle\rangle;\ !x)$ $\qquad p = \cdots$



# Island for Stack & Registers



# Garbage Collection

## Problem

$\ell \mapsto 3$

- if $\ell$ is collected ?
- if $\ell$ is relocated ?

## Solution

Global invariant:
all reachable memories in $M$
are live in $\Phi$

Logical Memory !
(allmost zero overhead)

$$W \dashv M \overset{repr}{\sim} \Phi$$

$$\downarrow \text{ execution}$$

$$W' \dashv M' \overset{repr}{\sim} \Phi'$$

$$\vdots \text{ alloc}$$

no gc in $M'$ $\longrightarrow$ $M' \uplus \square \overset{repr}{\sim} \Phi''$

# Language and World Specifications

$$\tau ::= \alpha \mid b \mid \tau_1 \times \tau_2 \mid \tau_1 \to \tau_2 \mid \forall \alpha.\, \tau \mid \exists \alpha.\, \tau \mid \mu \alpha.\, \tau \mid \mathsf{ref}\, \tau$$

$$\mathrm{CType} \stackrel{\text{def}}{=} \{ \tau \mid \mathrm{ftv}(\tau) = \emptyset \}$$

$$\begin{array}{l}
\mathrm{LangSpec} \stackrel{\text{def}}{=} \\
\{ (\mathrm{Val}, \mathrm{Com}, \mathrm{Cont}, \mathrm{Mem}, \mathrm{Conf}, \\
\quad \mathrm{plugv}, \mathrm{plugc}, \mathrm{step}, \mathrm{mdom}, \mathrm{mdisj}, \\
\quad \mathrm{oftype}, \mathrm{base}_b, \mathrm{pair}, \mathrm{app}, \mathrm{appty}, \\
\quad \mathrm{pack}, \mathrm{roll}, \mathrm{ref}, \mathrm{asgn}) \mid \\
\quad \mathrm{Val}, \mathrm{Com}, \mathrm{Cont}, \mathrm{Mem}, \mathrm{Conf} \in \mathrm{Set} \wedge \\
\quad \mathrm{plugv} \in \mathrm{Val} \times \mathrm{Cont} \times \mathrm{Mem} \to \mathbb{P}(\mathrm{Conf}) \wedge \\
\quad \mathrm{plugc} \in \mathrm{Com} \times \mathrm{Cont} \times \mathrm{Mem} \to \mathbb{P}(\mathrm{Conf}) \wedge \\
\quad \mathrm{step} \in \mathrm{Conf} \to \mathrm{Conf} \uplus \{ \mathit{fail}, \mathit{halt} \} \wedge \\
\quad \mathrm{mdom} \in \mathrm{Mem} \to \mathbb{P}(\mathrm{Val}) \wedge \\
\quad \mathrm{mdisj} \in \mathrm{Mem} \times \mathrm{Mem} \to \mathbb{P}(\mathrm{Mem}) \wedge \\
\quad \mathrm{oftype} \in \mathrm{CType} \to \mathbb{P}(\mathrm{Val} \times \mathrm{Mem}) \wedge \\
\quad \mathrm{base}_b \in [\![ b ]\!] \to \mathbb{P}(\mathrm{Val} \times \mathrm{Mem}) \wedge \\
\quad \mathrm{pair} \in \mathrm{Val} \times \mathrm{Val} \to \mathbb{P}(\mathrm{Val} \times \mathrm{Mem}) \wedge \\
\quad \mathrm{app} \in \mathrm{Val} \times \mathrm{Val} \to \mathbb{P}(\mathrm{Com}) \wedge \\
\quad \mathrm{appty} \in \mathrm{Val} \times \mathrm{CType} \to \mathbb{P}(\mathrm{Com}) \wedge \\
\quad \mathrm{pack} \in \mathrm{CType} \times \mathrm{Val} \to \mathbb{P}(\mathrm{Val} \times \mathrm{Mem}) \wedge \\
\quad \mathrm{roll} \in \mathrm{Val} \to \mathbb{P}(\mathrm{Val} \times \mathrm{Mem}) \wedge \\
\quad \mathrm{ref} \in \mathrm{Val} \to \mathbb{P}(\mathrm{Val} \times \mathrm{Mem}) \wedge \\
\quad \mathrm{asgn} \in \mathrm{Mem} \times \mathrm{Val} \times \mathrm{Val} \rightharpoonup \mathrm{Mem} \wedge \\
\quad \forall M_1, M_2.\, \forall M \in \mathrm{mdisj}(M_1, M_2). \\
\qquad \mathrm{mdom}(M) \supseteq \mathrm{mdom}(M_1) \uplus \mathrm{mdom}(M_2) \}
\end{array}$$

$$\begin{array}{l}
\text{For } \mathcal{L}_1, \mathcal{L}_2 \in \mathrm{LangSpec}, \\
\mathrm{WorldSpec} \stackrel{\text{def}}{=} \\
\{ (\mathrm{World}, \mathrm{lev}, \mathcal{M}, \mathcal{B}, \mathcal{O}, \triangleright, \sqsupseteq, \sqsupseteq_{\mathrm{pub}}) \mid \\
\quad \mathrm{World} \in \mathrm{Set} \wedge \\
\quad \mathrm{lev} \in \mathrm{World} \to \mathbb{N} \wedge \\
\quad \mathcal{M} \in \mathrm{World} \to \mathbb{P}(\mathcal{L}_1.\mathrm{Mem} \times \mathcal{L}_2.\mathrm{Mem}) \wedge \\
\quad \mathcal{B} \in \mathrm{World} \to \mathbb{P}(\mathcal{L}_1.\mathrm{Val} \times \mathcal{L}_2.\mathrm{Val}) \wedge \\
\quad \mathcal{O} \in \mathrm{World} \to \mathbb{P}(\mathcal{L}_1.\mathrm{Conf} \times \mathcal{L}_2.\mathrm{Conf}) \wedge \\
\quad \triangleright \in \mathrm{World} \to \mathrm{World} \wedge \\
\quad \sqsupseteq \in \mathbb{P}(\mathrm{World} \times \mathrm{World}) \wedge \\
\quad \sqsupseteq_{\mathrm{pub}} \in \mathbb{P}(\mathrm{World} \times \mathrm{World}) \wedge \\
\quad \sqsupseteq, \sqsupseteq_{\mathrm{pub}} \text{ are preorders} \wedge \sqsupseteq_{\mathrm{pub}} \subseteq \sqsupseteq \wedge \\
\quad \forall W' \sqsupseteq W.\, \triangleright W' \sqsupseteq \triangleright W \wedge \\
\quad \forall W' \sqsupseteq_{\mathrm{pub}} W.\, \triangleright W' \sqsupseteq_{\mathrm{pub}} \triangleright W \wedge \\
\quad \forall W.\, \triangleright W \sqsupseteq_{\mathrm{pub}} W \wedge \\
\quad \forall W' \sqsupseteq W.\, \mathrm{lev}(W') \le \mathrm{lev}(W) \wedge \\
\quad \forall W.\, \mathrm{lev}(W) > 0 \implies \mathrm{lev}(\triangleright W) = \mathrm{lev}(W) - 1 \}
\end{array}$$

$$
\begin{aligned}
\mathcal{V}[\![\alpha]\!]\rho &\stackrel{\text{def}}{=} \{\, (W, \mathbf{v}_1, v_2) \in \mathrm{oftype}(\alpha, \rho) \mid (W, \mathbf{v}_1, v_2) \in \Box\rho(\alpha).R \,\} \\
\mathcal{V}[\![b]\!]\rho &\stackrel{\text{def}}{=} \{\, (W, \mathbf{v}_1, v_2) \in \mathrm{oftype}(b, \rho) \mid \exists x \in [\![b]\!]\,. \\
&\qquad (W, \mathbf{v}_1, v_2) \in \Box\overline{(\mathcal{L}_1.\mathrm{base}_b(x), \mathcal{L}_2.\mathrm{base}_b(x))} \,\} \\
\mathcal{V}[\![\tau \times \tau']\!]\rho &\stackrel{\text{def}}{=} \{\, (W, \mathbf{v}_1, v_2) \in \mathrm{oftype}(\tau \times \tau', \rho) \mid \\
&\qquad \exists (\mathbf{u}_1, u_2) \in \triangleright\mathcal{V}[\![\tau]\!]\rho(W).\ \exists (\mathbf{u}_1', u_2') \in \triangleright\mathcal{V}[\![\tau']\!]\rho(W). \\
&\qquad (W, \mathbf{v}_1, v_2) \in \Box\overline{(\mathcal{L}_1.\mathrm{pair}(\mathbf{u}_1, \mathbf{u}_1'), \mathcal{L}_2.\mathrm{pair}(u_2, u_2'))} \,\} \\
\mathcal{V}[\![\tau' \to \tau]\!]\rho &\stackrel{\text{def}}{=} \{\, (W, \mathbf{v}_1, v_2) \in \mathrm{oftype}(\tau' \to \tau, \rho) \mid \forall W' \sqsupset_{\triangleright} W.\ \forall (\mathbf{u}_1, u_2) \in \mathcal{V}[\![\tau']\!]\rho(W'). \\
&\qquad \forall \mathbf{e}_1 \in \mathcal{L}_1.\mathrm{app}(\mathbf{v}_1, \mathbf{u}_1).\ \forall e_2 \in \mathcal{L}_2.\mathrm{app}(v_2, u_2).\ (W', \mathbf{e}_1, e_2) \in \mathcal{E}[\![\tau]\!]\rho \,\} \\
\mathcal{V}[\![\forall\alpha.\,\tau]\!]\rho &\stackrel{\text{def}}{=} \{\, (W, \mathbf{v}_1, v_2) \in \mathrm{oftype}(\forall\alpha.\,\tau, \rho) \mid \forall W' \sqsupset_{\triangleright} W.\ \forall (\tau_1, \tau_2, R) \in \mathrm{TyValRel}. \\
&\qquad \forall \mathbf{e}_1 \in \mathcal{L}_1.\mathrm{appty}(\mathbf{v}_1, \tau_1).\ \forall e_2 \in \mathcal{L}_2.\mathrm{appty}(v_2, \tau_2). \\
&\qquad (W', \mathbf{e}_1, e_2) \in \mathcal{E}[\![\tau]\!]\rho[\alpha \mapsto (\tau_1, \tau_2, R)] \,\} \\
\mathcal{V}[\![\exists\alpha.\,\tau]\!]\rho &\stackrel{\text{def}}{=} \{\, (W, \mathbf{v}_1, v_2) \in \mathrm{oftype}(\exists\alpha.\,\tau, \rho) \mid \\
&\qquad \exists (\tau_1, \tau_2, R) \in \mathrm{TyValRel}.\ \exists (\mathbf{u}_1, u_2) \in \mathcal{V}[\![\tau]\!]\rho[\alpha \mapsto (\tau_1, \tau_2, R)](W). \\
&\qquad (W, \mathbf{v}_1, v_2) \in \Box\overline{(\mathcal{L}_1.\mathrm{pack}(\tau_1, \mathbf{u}_1), \mathcal{L}_2.\mathrm{pack}(\tau_2, u_2))} \,\} \\
\mathcal{V}[\![\mathsf{ref}\,\tau]\!]\rho &\stackrel{\text{def}}{=} \{\, (W, \mathbf{v}_1, v_2) \in \mathrm{oftype}(\mathsf{ref}\,\tau, \rho) \mid \forall W' \sqsupseteq W.\ \forall (\mathbf{M}_1, M_2) \in \mathcal{M}(W'). \\
&\quad (\mathbf{v}_1, v_2) \in \mathcal{B}(W') \land \\
&\quad (\exists (\mathbf{u}_1, u_2) \in \triangleright\mathcal{V}[\![\tau]\!]\rho(W').\ (\mathbf{v}_1, \mathbf{M}_1) \in \mathcal{L}_1.\mathrm{ref}(\mathbf{u}_1) \land (v_2, M_2) \in \mathcal{L}_2.\mathrm{ref}(u_2)) \land \\
&\quad (\forall (\mathbf{u}_1, u_2) \in \triangleright\mathcal{V}[\![\tau]\!]\rho(W').\ (\mathcal{L}_1.\mathrm{asgn}(\mathbf{M}_1, \mathbf{v}_1, \mathbf{u}_1), \mathcal{L}_2.\mathrm{asgn}(M_2, v_2, u_2)) \in \mathcal{M}(W')) \,\} \\
\mathcal{V}[\![\mu\alpha.\,\tau]\!]\rho &\stackrel{\text{def}}{=} \mu(F_{\alpha,\tau,\rho}) \\
F_{\alpha,\tau,\rho} &\stackrel{\text{def}}{=} \lambda R.\ \{\, (W, \mathbf{v}_1, v_2) \in \mathrm{oftype}(\mu\alpha.\,\tau, \rho) \mid \\
&\qquad \exists (\mathbf{u}_1, u_2) \in \mathcal{V}[\![\tau]\!]\rho[\alpha \mapsto (\rho_1(\mu\alpha.\,\tau), \rho_2(\mu\alpha.\,\tau), R)](W). \\
&\qquad (W, \mathbf{v}_1, v_2) \in \Box\overline{(\mathcal{L}_1.\mathrm{roll}(\mathbf{u}_1), \mathcal{L}_2.\mathrm{roll}(u_2))} \,\} \\
\mu(F)(W) &\stackrel{\text{def}}{=} F(\mu(F)_{\sqsupset_{\triangleright} W})(W) \\
\mathcal{K}[\![\tau]\!]\rho &\stackrel{\text{def}}{=} \{\, (W, \mathbf{K}_1, K_2) \in \mathrm{World} \times \mathcal{L}_1.\mathrm{Cont} \times \mathcal{L}_2.\mathrm{Cont} \mid \forall W' \sqsupseteq_{\mathrm{pub}} W. \\
&\qquad \forall (\mathbf{v}_1, v_2) \in \mathcal{V}[\![\tau]\!]\rho(W').\ \forall (\mathbf{M}_1, M_2) \in \mathcal{M}(W'). \\
&\qquad \forall C_1 \in \mathcal{L}_1.\mathrm{plugv}(\mathbf{v}_1, \mathbf{K}_1, \mathbf{M}_1).\ \forall C_2 \in \mathcal{L}_2.\mathrm{plugv}(v_2, K_2, M_2). \\
&\qquad (C_1, C_2) \in \mathcal{O}(W') \,\} \\
\mathcal{E}[\![\tau]\!]\rho &\stackrel{\text{def}}{=} \{\, (W, \mathbf{e}_1, e_2) \in \mathrm{World} \times \mathcal{L}_1.\mathrm{Com} \times \mathcal{L}_2.\mathrm{Com} \mid \\
&\qquad \forall (\mathbf{K}_1, K_2) \in \mathcal{K}[\![\tau]\!]\rho(W).\ \forall (\mathbf{M}_1, M_2) \in \mathcal{M}(W). \\
&\qquad \forall C_1 \in \mathcal{L}_1.\mathrm{plugc}(\mathbf{e}_1, \mathbf{K}_1, \mathbf{M}_1).\ \forall C_2 \in \mathcal{L}_2.\mathrm{plugc}(e_2, K_2, M_2). \\
&\qquad (C_1, C_2) \in \mathcal{O}(W) \,\}
\end{aligned}
$$

$$
\begin{aligned}
\sqsupset_{\triangleright} &\stackrel{\text{def}}{=} \{\, (W', W) \mid \mathrm{lev}(W) > 0 \land W' \sqsupseteq \triangleright W \,\} \\
\mathrm{WVRel} &\stackrel{\text{def}}{=} \{\, R \in \mathbb{P}(\mathrm{World} \times \mathcal{L}_1.\mathrm{Val} \times \mathcal{L}_2.\mathrm{Val}) \,\} \\
R(W) &\stackrel{\text{def}}{=} \{\, (\mathbf{v}_1, v_2) \mid (W, \mathbf{v}_1, v_2) \in R \,\} \\
\triangleright R &\stackrel{\text{def}}{=} \{\, (W, \mathbf{v}_1, v_2) \mid \mathrm{lev}(W) > 0 \implies (\triangleright W, \mathbf{v}_1, v_2) \in R \,\} \\
\Box R &\stackrel{\text{def}}{=} \{\, (W, \mathbf{v}_1, v_2) \mid \forall W' \sqsupseteq W.\ (W', \mathbf{v}_1, v_2) \in R \,\} \\
R_{\sqsupset_{\triangleright} W} &\stackrel{\text{def}}{=} \{\, (W', \mathbf{v}_1, v_2) \mid W' \sqsupset_{\triangleright} W \land (W', \mathbf{v}_1, v_2) \in R \,\} \\
\overline{(R_1, R_2)} &\stackrel{\text{def}}{=} \{\, (W, \mathbf{v}_1, v_2) \mid \forall (\mathbf{M}_1, M_2) \in \mathcal{M}(W).\ (\mathbf{v}_1, \mathbf{M}_1) \in R_1 \land (v_2, M_2) \in R_2 \,\} \\
&\quad \text{for } R_1 \in \mathbb{P}(\mathcal{L}_1.\mathrm{Val} \times \mathcal{L}_1.\mathrm{Mem}),\ R_2 \in \mathbb{P}(\mathcal{L}_2.\mathrm{Val} \times \mathcal{L}_2.\mathrm{Mem}) \\
\mathrm{TyValRel} &\stackrel{\text{def}}{=} \{\, (\tau_1, \tau_2, R) \mid \tau_1, \tau_2 \in \mathrm{CType} \land R \in \mathrm{WVRel} \,\} \\
\rho &\in \mathrm{TypeVar} \rightharpoonup \mathrm{TyValRel} \\
\rho_1(\tau) &\stackrel{\text{def}}{=} \tau[\rho(\alpha).\tau_1/\alpha] \qquad \rho_2(\tau) \stackrel{\text{def}}{=} \tau[\rho(\alpha).\tau_2/\alpha] \\
\mathrm{oftype}(\tau, \rho) &\stackrel{\text{def}}{=} \Box\overline{(\mathcal{L}_1.\mathrm{oftype}(\rho_1(\tau)), \mathcal{L}_2.\mathrm{oftype}(\rho_2(\tau)))}
\end{aligned}
$$

Logical relation

Thm (Monotonicity)

$v_1 \approx_\tau v_2 : W$

$\Rightarrow \forall W' \sqsupseteq W.$

$v_1 \approx_\tau v_2 : W'$

# Logical Relation : details

$$\mathcal{V}[\![\tau' \to \tau]\!]\rho \stackrel{\text{def}}{=} \{\, (W, \mathbf{v}_1, v_2) \in \text{oftype}(\tau' \to \tau, \rho) \mid \forall W' \sqsupset_{\triangleright} W.\ \forall(\mathbf{u}_1, u_2) \in \mathcal{V}[\![\tau']\!]\rho(W')$$
$$\forall \mathbf{e}_1 \in \mathcal{L}_1.\text{app}(\mathbf{v}_1, \mathbf{u}_1).\ \forall e_2 \in \mathcal{L}_2.\text{app}(v_2, u_2).\ (W', \mathbf{e}_1, e_2) \in \mathcal{E}[\![\tau]\!]\rho \,\}$$

$$\mathcal{V}[\![\mathsf{ref}\ \tau]\!]\rho \stackrel{\text{def}}{=} \{\, (W, \mathbf{v}_1, v_2) \in \text{oftype}(\mathsf{ref}\ \tau, \rho) \mid \forall W' \sqsupseteq W.\ \forall(\mathbf{M}_1, M_2) \in \mathcal{M}(W').$$
$$(\mathbf{v}_1, v_2) \in \mathcal{B}(W') \land$$
$$(\exists(\mathbf{u}_1, u_2) \in \triangleright\mathcal{V}[\![\tau]\!]\rho(W').\ (\mathbf{v}_1, \mathbf{M}_1) \in \mathcal{L}_1.\text{ref}(\mathbf{u}_1) \land (v_2, M_2) \in \mathcal{L}_2.\text{ref}(u_2)) \land$$
$$(\forall(\mathbf{u}_1, u_2) \in \triangleright\mathcal{V}[\![\tau]\!]\rho(W').\ (\mathcal{L}_1.\text{asgn}(\mathbf{M}_1, \mathbf{v}_1, \mathbf{u}_1), \mathcal{L}_2.\text{asgn}(M_2, v_2, u_2)) \in \mathcal{M}(W'))\}$$

supports both ( iso-recursive types ⇝ High
equi-recursive types ⇝ Low

# language-generic world model

World : LangSpec × LangSpec → WorldSpec

↳ a functor implementing

Dreyer et al.'s possible worlds model

It can be built on arbitrary LangSpecs!

Low Spec

$$
\begin{array}{lll}
\mathrm{Loc} & \stackrel{\mathrm{def}}{=} & \{\, \boldsymbol{l} \in \mathbb{N} \,\} \\
\mathrm{Word} & \stackrel{\mathrm{def}}{=} & \{\, w \in \mathbb{N} \,\} \\
\mathbf{v} \in \mathrm{Val} & ::= & \underline{w} \mid \widehat{\boldsymbol{l}} \\
\mathrm{lv} \in \mathrm{Lvalue} & ::= & \lfloor r \rfloor \mid \langle a \rangle_{\mathrm{s}} \mid \langle r - o \rangle_{\mathrm{s}} \mid \langle \boldsymbol{l} : o \rangle_{\mathrm{h}} \mid \langle r + o \rangle_{\mathrm{h}} \\
\mathrm{rv} \in \mathrm{Rvalue} & ::= & \mathrm{lv} \mid \mathbf{v} \\
\mathrm{Com} & \stackrel{\mathrm{def}}{=} & \{\, \mathbf{e} = (\mathrm{cpc}, \mathrm{kpc}, \mathrm{vloc}, \mathrm{data}) \\
 & & \quad \in \mathrm{Rvalue} \times \mathrm{Rvalue} \times \mathrm{Lvalue} \times \mathbb{P}(\mathrm{Mem}) \,\} \\
\mathrm{Cont} & \stackrel{\mathrm{def}}{=} & \{\, \mathbf{K} = (\mathrm{kpc}, \mathrm{vloc}) \in \mathrm{PAddr} \times \mathrm{Lvalue} \,\} \\
\mathrm{CodeFrag} & \stackrel{\mathrm{def}}{=} & \mathrm{PAddr} \rightharpoonup_{\mathrm{fin}} \mathrm{Instruction} \\
\mathrm{RegFiles} & \stackrel{\mathrm{def}}{=} & (\mathrm{Register} \setminus \{\mathrm{sp}\} \to \mathrm{Val}) \uplus \{\, \mathrm{undef} \,\} \\
\mathrm{List}\ X & \stackrel{\mathrm{def}}{=} & \{\, (x_0, \ldots, x_{n-1}) \mid n \in \mathbb{N} \wedge x_0, \ldots, x_{n-1} \in X \,\} \\
\mathrm{Stack} & \stackrel{\mathrm{def}}{=} & \mathrm{List}\ \mathrm{Val} \uplus \{\, \mathrm{undef} \,\} \\
\mathrm{Heap} & \stackrel{\mathrm{def}}{=} & \mathrm{Loc} \rightharpoonup_{\mathrm{fin}} \mathrm{List}\ \mathrm{Val} \\
\mathrm{Table} & \stackrel{\mathrm{def}}{=} & (\mathrm{Loc} \to \mathbb{N} \times \mathrm{PAddr}) \uplus \{\, \mathrm{undef} \,\} \\
\mathrm{SysHeap} & \stackrel{\mathrm{def}}{=} & (\mathrm{PAddr} \rightharpoonup \mathrm{Word}) \uplus \{\, \mathrm{undef} \,\} \\
\mathrm{Mem} & \stackrel{\mathrm{def}}{=} & \{\, \mathbf{M} = (\mathrm{code}, \mathrm{reg}, \mathrm{stk}, \mathrm{hp}, \mathrm{tab}, \mathrm{shp}) \\
 & & \in \mathrm{CodeFrag} \times \mathrm{RegFiles} \times \mathrm{Stack} \times \mathrm{Heap} \times \mathrm{Table} \times \mathrm{SysHeap} \,\} \\
\mathrm{Conf} & \stackrel{\mathrm{def}}{=} & \mathrm{PConf}
\end{array}
$$

$$
\begin{aligned}
\mathrm{plugv}(\mathbf{v}, \mathbf{K}, \mathbf{M}) \stackrel{\mathrm{def}}{=} \{\, & (\Phi, \mathrm{pc}) \in \mathrm{Conf} \mid \mathbf{M} \text{ repr } \Phi \wedge \\
& \mathrm{pc} = \mathbf{K}.\mathrm{kpc} \wedge \mathbf{M}(\mathbf{K}.\mathrm{vloc}) = \mathbf{v} \,\} \\
\mathrm{plugc}(\mathbf{e}, \mathbf{K}, \mathbf{M}) \stackrel{\mathrm{def}}{=} \{\, & (\Phi, \mathrm{pc}) \in \mathrm{Conf} \mid \mathbf{M} \text{ repr } \Phi \wedge \mathbf{M} \in \mathbf{e}.\mathrm{data} \wedge \\
& \mathrm{pc} = \mathbf{M}(\mathbf{e}.\mathrm{cpc}) \wedge \mathbf{M}(\mathbf{e}.\mathrm{kpc}) = \underline{\mathbf{K}.\mathrm{kpc}} \wedge \mathbf{e}.\mathrm{vloc} = \mathbf{K}.\mathrm{vloc} \,\}
\end{aligned}
$$

$$
\begin{aligned}
\mathrm{oftype}(\tau) \;&\stackrel{\text{def}}{=}\; \{\, (\mathbf{v}, \mathbf{M}) \in \mathrm{Val} \times \mathrm{Mem} \mid \\
&\qquad \forall \tau_1, \tau_2.\, \tau = \tau_1 \to \tau_2 \implies \exists \boldsymbol{l}, w.\, \mathbf{v} = \widehat{\boldsymbol{l}} \wedge \mathbf{M}.\mathrm{hp}(\boldsymbol{l})(0) = \underline{w} \wedge \\
&\qquad\quad \forall \alpha, \tau'.\, \tau = \forall \alpha.\, \tau' \implies \exists \boldsymbol{l}, w.\, \mathbf{v} = \widehat{\boldsymbol{l}} \wedge \mathbf{M}.\mathrm{hp}(\boldsymbol{l})(0) = \underline{w} \,\} \\
\mathrm{base}_b(x) \;&\stackrel{\text{def}}{=}\; \{\, (\mathbf{v}, \mathbf{M}) \in \mathrm{Val} \times \mathrm{Mem} \mid \mathbf{v} \text{ is a representation of } x \,\} \\
\mathrm{pair}(\mathbf{v}_1, \mathbf{v}_2) \;&\stackrel{\text{def}}{=}\; \{\, (\mathbf{v}, \mathbf{M}) \in \mathrm{Val} \times \mathrm{Mem} \mid \exists \boldsymbol{l}.\, \mathbf{v} = \widehat{\boldsymbol{l}} \wedge \\
&\qquad \mathbf{M}.\mathrm{hp}(\boldsymbol{l})(0) = \mathbf{v}_1 \wedge \mathbf{M}.\mathrm{hp}(\boldsymbol{l})(1) = \mathbf{v}_2 \,\} \\
\mathrm{app}(\mathbf{v}_1, \mathbf{v}_2) \;&\stackrel{\text{def}}{=}\; \{\, \mathbf{e} \in \mathrm{Com} \mid \exists \boldsymbol{l}.\, \mathbf{v}_1 = \widehat{\boldsymbol{l}} \wedge \\
&\quad \mathbf{e}.\mathrm{cpc} = \langle \boldsymbol{l} : 0 \rangle_{\mathrm{h}} \wedge \mathbf{e}.\mathrm{kpc} = \lfloor \mathrm{wk}_0 \rfloor \wedge \mathbf{e}.\mathrm{vloc} = \lfloor \mathrm{wk}_5 \rfloor \wedge \\
&\quad \mathbf{e}.\mathrm{data} = \{\, \mathbf{M} \in \mathrm{Mem} \mid \mathbf{M}.\mathrm{reg}(\mathrm{wk}_1) = \mathbf{v}_1 \wedge \mathbf{M}.\mathrm{reg}(\mathrm{wk}_2) = \mathbf{v}_2 \,\} \,\} \\
\mathrm{appty}(\mathbf{v}, \tau) \;&\stackrel{\text{def}}{=}\; \{\, \mathbf{e} \in \mathrm{Com} \mid \exists \boldsymbol{l}.\, \mathbf{v} = \widehat{\boldsymbol{l}} \wedge \\
&\qquad \mathbf{e}.\mathrm{cpc} = \langle \boldsymbol{l} : 0 \rangle_{\mathrm{h}} \wedge \mathbf{e}.\mathrm{kpc} = \lfloor \mathrm{wk}_0 \rfloor \wedge \mathbf{e}.\mathrm{vloc} = \lfloor \mathrm{wk}_5 \rfloor \wedge \\
&\qquad \mathbf{e}.\mathrm{data} = \{\, \mathbf{M} \in \mathrm{Mem} \mid \mathbf{M}.\mathrm{reg}(\mathrm{wk}_1) = \mathbf{v} \,\} \,\} \\
\mathrm{pack}(\tau, \mathbf{v}) \;&\stackrel{\text{def}}{=}\; \{\, (\mathbf{v}', \mathbf{M}) \in \mathrm{Val} \times \mathrm{Mem} \mid \mathbf{v}' = \mathbf{v} \,\} \\
\mathrm{roll}(\mathbf{v}) \;&\stackrel{\text{def}}{=}\; \{\, (\mathbf{v}', \mathbf{M}) \in \mathrm{Val} \times \mathrm{Mem} \mid \mathbf{v}' = \mathbf{v} \,\} \\
\mathrm{ref}(\mathbf{v}) \;&\stackrel{\text{def}}{=}\; \{\, (\mathbf{v}', \mathbf{M}) \in \mathrm{Val} \times \mathrm{Mem} \mid \exists \boldsymbol{l}.\, \mathbf{v}' = \widehat{\boldsymbol{l}} \wedge \mathbf{M}.\mathrm{hp}(\boldsymbol{l})(0) = \mathbf{v} \,\} \\
\mathrm{asgn}(\mathbf{M}, \mathbf{v}_1, \mathbf{v}_2) \;&\stackrel{\text{def}}{=}\; \begin{cases} \mathbf{M}[\boldsymbol{l} : 0 \mapsto \mathbf{v}_2]_{\mathrm{hp}} & \text{if } \mathbf{v}_1 = \widehat{\boldsymbol{l}} \wedge |\mathbf{M}.\mathrm{hp}(\boldsymbol{l})| > 0 \\ \mathrm{undef} & \text{otherwise} \end{cases}
\end{aligned}
$$

# Specification of GC

$$
\begin{aligned}
&\mathbf{v}\ \text{live in}\ \mathbf{M} \overset{\text{def}}{=} \begin{cases} \top & \text{if } \mathbf{v} = \underline{w} \\ \exists n, a.\ \mathbf{M}.\text{tab}(l) = (n, a) \wedge n > 0 & \text{if } \mathbf{v} = \widehat{l} \end{cases} \\
&\text{reach}_0(\mathbf{M}) \overset{\text{def}}{=} \{\, l \mid \exists r \in \text{Register}.\ \widehat{l} = \mathbf{M}.\text{reg}(r) \,\} \cup \\
&\qquad\qquad \{\, l \mid \exists j < |\mathbf{M}.\text{stk}|.\ \widehat{l} = \mathbf{M}.\text{stk}(j) \,\} \\
&\text{reach}_{i+1}(\mathbf{M}) \overset{\text{def}}{=} \text{reach}_i(\mathbf{M}) \cup \\
&\qquad\qquad \{\, l \mid \exists l' \in \text{reach}_i(\mathbf{M}).\ \exists j.\ \widehat{l} = \mathbf{M}.\text{hp}(l')(j) \,\} \\
&\text{reach}(\mathbf{M}) \overset{\text{def}}{=} \textstyle\bigcup_{i \in \mathbb{N}} \text{reach}_i(\mathbf{M}) \\
&\text{GCSpec} \overset{\text{def}}{=} \\
&\{\, \mathcal{G} \in \text{PAddr} \to \{\, (\text{init}, \text{alloc}, \text{code}, I) \\
&\qquad\qquad \in \text{PAddr} \times \text{PAddr} \times \text{List Instruction} \times \\
&\qquad\qquad\quad \mathbb{P}(\text{Table} \times \text{SysHeap}) \,\} \mid \\
&\ \forall \text{gcbg}, \Phi, \text{pc}. \\
&\quad \Phi.\text{code} \supseteq [\text{gcbg} \mapsto \mathcal{G}(\text{gcbg}).\text{code}] \wedge \Phi.\text{reg}(\text{wk}_4) = \underline{\text{pc}} \implies \\
&\quad \exists \mathbf{M}', \Phi'. \\
&\quad\ (\Phi, \mathcal{G}(\text{gcbg}).\text{init}) \overset{*}{\hookrightarrow} (\Phi', \text{pc}) \wedge \\
&\quad\ \Phi'.\text{code} = \Phi.\text{code} \wedge \mathbf{M}'.\text{code} = [\text{gcbg} \mapsto \mathcal{G}(\text{gcbg}).\text{code}] \wedge \\
&\quad\ \mathbf{M}'\ \text{repr}\ \Phi' \wedge \mathbf{M}' \in \mathcal{G}.\mathit{GR}(\text{gcbg}) \wedge \mathbf{M}' \in \mathcal{G}.\mathit{MR}(\text{gcbg}) \wedge \\
&\ \forall \text{gcbg}, \mathbf{M}, \Phi, \text{pc}, n. \\
&\quad \mathbf{M}\ \text{repr}\ \Phi \wedge \mathbf{M} \in \mathcal{G}.\mathit{GR}(\text{gcbg}) \wedge \mathbf{M} \in \mathcal{G}.\mathit{MR}(\text{gcbg}) \wedge \\
&\quad \mathbf{M}.\text{reg}(\text{wk}_4) = \underline{\text{pc}} \wedge \mathbf{M}.\text{reg}(\text{wk}_5) = \underline{n} \implies \\
&\quad \exists \Phi', \mathbf{M}', T, S, w, l, w_0, \ldots, w_{n-1}. \\
&\quad\ (\Phi, \mathcal{G}(\text{gcbg}).\text{alloc}) \overset{*}{\hookrightarrow} (\Phi', \text{pc}) \wedge \\
&\quad\ \mathbf{M}'\ \text{repr}\ \Phi' \wedge \mathbf{M}' \in \mathcal{G}.\mathit{GR}(\text{gcbg}) \wedge \mathbf{M}' \in \mathcal{G}.\mathit{MR}(\text{gcbg}) \wedge \\
&\quad\ \mathbf{M}' = \mathbf{M}[[T, S]][\text{wk}_4 \mapsto \underline{w}]_{\text{reg}}[\text{wk}_5 \mapsto \widehat{l}\,]_{\text{reg}} \uplus \\
&\qquad\qquad [l \mapsto (\underline{w_0}, \ldots, \underline{w_{n-1}})]_{\text{hp}} \,\} \\
&\mathcal{G}.\mathit{GR}(\text{gcbg}) \overset{\text{def}}{=} \{\, \mathbf{M} \in \text{Mem} \mid \forall l \in \text{reach}(\mathbf{M}).\ \widehat{l}\ \text{live in}\ \mathbf{M} \,\} \\
&\mathcal{G}.\mathit{MR}(\text{gcbg}) \overset{\text{def}}{=} \{\, \mathbf{M} \in \text{Mem} \mid (\mathbf{M}.\text{tab}, \mathbf{M}.\text{shp}) \in \mathcal{G}(\text{gcbg}).I \wedge \\
&\qquad\qquad \mathbf{M}.\text{code} \supseteq [\text{gcbg} \mapsto \mathcal{G}(\text{gcbg}).\text{code}] \,\}
\end{aligned}
$$

Mark-Sweep & Copying GC satisy

Spec of init

Spec of alloc

→ all reachable memories are live

→ private invariant of GC

## Program Equivalence

$$\mathcal{H}.\mathrm{Prog} \stackrel{\mathrm{def}}{=} \{\, e \mid \mathrm{floc}(e) = \emptyset \,\}$$
$$\mathcal{L}.\mathrm{Prog} \stackrel{\mathrm{def}}{=} \{\, p \in \mathrm{PAddr} \times \mathrm{PAddr} \to \mathrm{List\ Instruction} \,\}$$

$$\mathcal{D}[\![\cdot]\!] \stackrel{\mathrm{def}}{=} \emptyset$$
$$\mathcal{D}[\![\Delta, \alpha]\!] \stackrel{\mathrm{def}}{=} \{\, (\rho, \alpha \mapsto R) \mid \rho \in \mathcal{D}[\![\Delta]\!] \wedge R \in \mathrm{TyValRel} \,\}$$

$$\mathcal{G}[\![\cdot]\!]\rho \stackrel{\mathrm{def}}{=} \{\, (W, \mathbf{v}, \emptyset) \mid W \in \mathrm{World} \wedge \mathbf{v} \in \mathcal{L}.\mathrm{Val} \,\}$$
$$\begin{aligned}\mathcal{G}[\![\Gamma, x : \tau]\!]\rho \stackrel{\mathrm{def}}{=} \{\, (W, \mathbf{v}, (\gamma, x \mapsto v)) \mid{}& \exists \mathbf{v}_1, \mathbf{v}_2 .\\ & (W, \mathbf{v}, \langle\rangle) \in \square\overline{(\mathcal{L}.\mathrm{pair}(\mathbf{v}_1, \mathbf{v}_2), \mathcal{H}.\mathrm{Val} \times \mathcal{H}.\mathrm{Mem})} \wedge \\ & (W, \mathbf{v}_1, v) \in \mathcal{V}[\![\tau]\!]\rho \wedge (W, \mathbf{v}_2, \gamma) \in \mathcal{G}[\![\Gamma]\!]\rho \,\}\end{aligned}$$

$$W_k^\circ(\mathcal{G}, \mathrm{gcbg}) \stackrel{\mathrm{def}}{=} (k, [\iota^{\mathrm{regstk}}, \iota^{\mathrm{htyping}}, \iota^{\mathrm{gc}}(\mathcal{G}, \mathrm{gcbg})], GR^\circ(\mathcal{G}, \mathrm{gcbg}))$$

$$\Delta ; \Gamma \vdash p \approx e : \tau \stackrel{\mathrm{def}}{=}$$

$$\begin{aligned}&\emptyset ; \Delta ; \Gamma \vdash e : \tau \wedge \\ &\forall \mathcal{G}, \mathrm{gcbg}, \mathrm{bg}.\ \forall k, W \sqsupseteq W_k^\circ(\mathcal{G}, \mathrm{gcbg}).\ \forall (\mathbf{M}, M) \in \mathcal{M}(W). \\ &\forall \mathbf{M}'.\ \mathbf{M}' = \mathbf{M} \uplus [\mathrm{bg} \mapsto p(\mathcal{G}(\mathrm{gcbg}).\mathrm{alloc}, \mathrm{bg})]_{\mathrm{code}} \implies \\ &\exists W' \sqsupseteq W.\ \mathrm{lev}(W') = \mathrm{lev}(W) \wedge (\mathbf{M}', M) \in \mathcal{M}(W') \wedge \\ &\quad \forall W'' \sqsupseteq W'.\ \forall \rho \in \mathcal{D}[\![\Delta]\!] .\ \forall (\mathbf{v}, \gamma) \in \mathcal{G}[\![\Gamma]\!]\rho(W''). \\ &\qquad ((\underline{\mathrm{bg}}, \lfloor \mathrm{wk}_0 \rfloor, \lfloor \mathrm{wk}_5 \rfloor, \{\, \mathbf{M} \mid \mathbf{M}.\mathrm{reg}(\mathrm{sv}_0) = \mathbf{v} \,\}), \gamma\rho e) \in \mathcal{E}[\![\tau]\!]\rho(W'') \\ &\text{where } \gamma\rho e ::= e[\rho(\alpha).\tau_2/\alpha][\gamma(x)/x] .\end{aligned}$$

# Results

$$\Delta; \Gamma \vdash p \approx e : \tau$$

1. Adequacy

2. Compositionality

3. Compiler Correctness for a simple compiler

# Summary

Summary

- Language-generic logical relation
- Adq & Compositional relation between Low & High
- use of Logical Memory for G.C.
- Compositional Compiler Correctness

Comments : Low-Low, High-High relations : no problem!
Generational GC : no problem!

## Future work

- Multi-phase compiler
- Coq formalization
- Concurrency
- Self-modifing code applications:
  dynamic linking & loading ; dynamic code generation
  JIT comoiler
- input, output
- other languages
  Assem-C , C-ML

Thank you !!