

$$\begin{aligned}
c &::= \text{skip} \mid \text{let } x : s := e \text{ in } c \mid \text{if } e \text{ c } c \mid \text{while } e \text{ c } c \mid x := e \mid *e := e \\
&\mid x := *e \mid \text{at } k \text{ e } c \mid \text{if } (k \sqsubseteq k) \text{ c } c \mid \text{match } \alpha \ \bar{p} \Rightarrow \bar{c} \mid x := \text{fp} \mid f(\bar{k})(\bar{s})(\bar{e}) \\
&\mid \text{let } (\alpha : \text{type}_k, x : s) := e \text{ in } c \mid \text{let } (\kappa : \text{level}_k, x : s) := e \text{ in } c \\
e &::= n \mid x \mid e \oplus e \mid \text{null} \mid \text{unroll } e \mid \text{roll } e \mid \text{pack } (s, e) \text{ as } \exists \alpha : \text{type}_k. s \mid \text{sizeof } s \\
&\mid \text{pack } (k, e) \text{ as } \exists \kappa : \text{level}_k. s \mid \&x \\
k &::= \ell \mid \kappa \mid k \sqcup k \mid k \sqcap k \\
s &::= t_k \mid \alpha^k \mid \bar{s} \\
t &::= \text{int} \mid k \mapsto s \mid s @ s \mid \exists \alpha : \text{type}_k. s \\
&\mid \exists \kappa : \text{level}_k. s \mid \mu \alpha : \text{type}_k. s \mid \text{size}[s] \\
p &::= \text{int}_\kappa \mid (p @ p)_\kappa \mid (\kappa \mapsto p)_\kappa \mid \bar{p} \mid \alpha
\end{aligned}$$

Fig. 21: The syntax of Zee. We write α to mean α^\perp .

$$\begin{aligned}
v &::= n \mid a_\nu \mid (\ell, v) \mid \text{fp} \mid (\tau, v) \\
\tau &::= \pi_\ell \mid \bar{\tau} \mid \frac{1}{2} \\
\pi &::= \text{int} \mid \ell \mapsto \tau \mid \tau @ \tau \mid \exists \alpha : \text{type}. s \\
&\mid \exists \kappa : \text{level}. s \mid \mu \alpha : \text{type}. s \mid \text{size}[\tau]
\end{aligned}$$

Fig. 22: Values in Zee.

a) *Commands and expressions*: Commands are ranged over by the meta-variable c , and expressions are ranged over by the meta-variable e . Figure 21 shows the syntax of commands and expressions.

b) *Labels and security types*: Labels are ranged over by k and include literals ℓ from some fixed lattice \mathcal{L} , variables. Security types are ranged over by the meta-variable s . We include a security label k when definition security type variables α (i.e., we write α^k instead of just α) to properly define the notion of raising a security type to the label of a security label.

c) *Base types*: Base types are ranged over by the meta-variable s .

$$\begin{aligned}
\mathcal{F} &::= f(\overline{\kappa : \text{level}_{k_1}})(\overline{\alpha : \text{type}_{k_2}})(\overline{x : s}) =_{pc}^{fr} c \\
\mathcal{P} &::= \bar{\mathcal{F}}; c
\end{aligned}$$

Relation $\tau \lesssim p$ specifies that the fully evaluated type τ *matches* pattern p . This is needed to define the semantics of pattern matching.

$$\begin{array}{c}
\boxed{s \lesssim p} \\
\hline
\overline{\text{int}_\ell \lesssim \text{int}_\kappa} \quad \overline{\tau \lesssim p} \quad \overline{\tau_i \lesssim p_i \ i = 1, 2} \quad \overline{\tau \lesssim \alpha} \\
\hline
\overline{(\ell_1 \mapsto \tau)_{\ell_2} \lesssim (\kappa_1 \mapsto p)_{\kappa_2}} \quad \overline{(\tau_1 @ \tau_2)_\ell \lesssim (p_1 @ p_2)_\kappa} \\
\hline
\overline{\begin{array}{c} |\bar{\tau}| = n \quad |\bar{p}| = m \quad m \leq n \\ \forall i \in \{1, \dots, m-1\} . \tau_i \lesssim p_i \quad \tau_{m \dots n} \lesssim p_m \end{array}} \\
\hline
\overline{\bar{\tau} \lesssim \bar{p}}
\end{array}$$

When an evaluated type matches a pattern, the pattern initializes the free variables based on the evaluated type. The interpretation of p , written $\llbracket p \rrbracket$, is a function that receives the store, the frame, and the matched type, and returns an updated store and an updated frame. The frame is updated to keep the type information in the frame up-to-date with the local variables from in the pattern being matched.

$$\boxed{\llbracket p \rrbracket(p, \tau) = p'}$$

$$\begin{array}{c}
\text{S-IF-T} \quad \frac{\langle e, M, P \rangle \Downarrow n \quad n \neq 0}{\langle \text{if } e \ c_1 \ c_2, M, P, q \rangle_\nu \rightarrow \langle c_1, M, P, q+1 \rangle_\nu} \\
\text{S-IF-F} \quad \frac{\langle e, M, P \rangle \Downarrow 0}{\langle \text{if } e \ c_1 \ c_2, M, P, q \rangle_\nu \rightarrow \langle c_2, M, P, q+1 \rangle_\nu} \\
\text{S-WHILE-F} \quad \frac{\langle e, M, P \rangle \Downarrow 0}{\langle \text{while } e \ c, M, P, q \rangle_\nu \rightarrow \langle \text{stop}, M, P, q+1 \rangle_\nu} \\
\text{S-WHILE-T} \quad \frac{\langle e, M, P \rangle \Downarrow n \quad n \neq 0}{\langle \text{while } e \ c, M, P, q \rangle_\nu \rightarrow \langle c; \text{while } e \ c, M, P, q+1 \rangle_\nu} \\
\text{S-SKIP} \quad \langle \text{skip}, M, P, q \rangle_\nu \rightarrow \langle \text{stop}, M, P, q+1 \rangle_\nu \\
\text{S-ASGN} \quad \frac{\langle e, m \cdot M, P \rangle \Downarrow v \quad m' = m[\delta(x) + \text{fp}(m) \mapsto v]}{\langle x := e, m \cdot M, P, q \rangle_\nu \rightarrow \langle \text{stop}, m' \cdot M, P, q+1 \rangle_\nu} \\
\text{S-WRITE} \quad \frac{m_i = (m_{\mathbb{I}}, \nu_i) \in M \quad a \in \mathbb{I} \quad \nu_i \leq \gamma \quad \langle e_1, M, P \rangle \Downarrow a_\gamma \quad \langle e_2, M, P \rangle \Downarrow v}{\langle *e_1 := e_2, M, P, q \rangle_\nu \rightarrow \langle \text{stop}, M[a \mapsto v], P, q+1 \rangle_\nu} \\
\text{S-READ} \quad \frac{M = m \cdot M' \quad m_i = (m_{\mathbb{I}}, \nu_i) \in M \quad a \in \mathbb{I} \quad \nu_i \leq \gamma \quad \langle e, M, P \rangle \Downarrow a_\gamma \quad m' = m[\delta(x) + \text{fp}(m) \mapsto M(n)]}{\langle x := *e, m \cdot M, P, q \rangle_\nu \rightarrow \langle \text{stop}, m' \cdot M', P, q+1 \rangle_\nu} \\
\text{S-AT} \quad \frac{\langle e, m, P \rangle \Downarrow n}{\langle \text{at } k \ e \ c, m, P, q \rangle_\nu \rightarrow \langle c; \text{delay } n, m, P, q+1 \rangle_\nu} \\
\text{S-DELAY} \quad \frac{n \leq q}{\langle \text{delay } n, m, P, q \rangle_\nu \rightarrow \langle \text{delay } n, m, P, n+1 \rangle_\nu} \\
\text{S-LET} \quad \frac{M = m \cdot M' \quad \langle s, p \rangle \Downarrow_{\text{sectype } \tau} \tau \quad \langle e, m, p \rangle \Downarrow v \quad m' = m[\delta(x) + \text{fp}(m) \mapsto v] \quad p' = p[p_{\text{local}} \mapsto p_{\text{local}}[x \mapsto \tau]]}{\langle \text{let } x : s := e \text{ in } c, M, p \cdot P, q \rangle_\nu \rightarrow \langle c; \text{unscope}(x), m' \cdot M', p' \cdot P, q+1 \rangle_\nu} \\
\text{S-UNSCOPE} \quad \frac{p' = p[p_{\text{local}} \mapsto p_{\text{local}}[x \mapsto \perp]]}{\langle \text{unscope}(x), M, p \cdot P, q \rangle_\nu \rightarrow \langle \text{stop}, M, p' \cdot P, q+1 \rangle_\nu} \\
\text{S-UNPACK-LEV} \quad \frac{P = p \cdot P' \quad \langle s, p' \rangle \Downarrow_{\text{sectype } \tau} \tau \quad \langle e, m, p \rangle \Downarrow (\ell_1, v_2) \quad p' = p[p_{\text{var}} \mapsto p_{\text{var}}[\kappa \mapsto \ell_1], p_{\text{local}} \mapsto p_{\text{local}}[x \mapsto \tau]] \quad M = m \cdot M' \quad m' = m[\delta(x) + \text{fp}(m) \mapsto v_2]}{\langle \text{let } (\kappa : \text{level}_k, x : s) := e \text{ in } c, M, P, q \rangle_\nu \rightarrow \langle c; \text{unscope}(x), m' \cdot M', p' \cdot P', q+1 \rangle_\nu} \\
\text{S-UNPACK-TY} \quad \frac{P = p \cdot P' \quad \langle s, p' \rangle \Downarrow_{\text{sectype } \tau} \tau \quad \langle e, m, p \rangle \Downarrow (\tau_1, v_2) \quad p' = p[p_{\text{var}} \mapsto p_{\text{var}}[\alpha \mapsto \tau_1], p_{\text{local}} \mapsto p_{\text{local}}[x \mapsto \tau]] \quad M = m \cdot M' \quad m' = m[\delta(x) + \text{fp}(m) \mapsto v_2]}{\langle \text{let } (\alpha : \text{type}_k, x : s) := e \text{ in } c, M, P, q \rangle_\nu \rightarrow \langle c; \text{unscope}(x), m' \cdot M', p' \cdot P', q+1 \rangle_\nu} \\
\text{S-EPILOGUE} \quad \frac{}{\langle \text{epilogue}, (\mathbb{I}_1, |m_1|, \nu_1) \cdot (\mathbb{I}_2, |m_2|, \nu_2) \cdot M, p \cdot P, q \rangle_\nu \rightarrow \langle \text{stop}, (\mathbb{I}_2, |m_2|, \max(\nu_1, \nu_2) + 1) \cdot M, P, q+1 \rangle_\nu} \\
\text{S-MATCH} \quad \frac{\text{argmin}_{i=1, \dots, n}(\tau \preceq p_i) = j \quad \langle \alpha, p \rangle \Downarrow_{\text{sectype } \tau} \tau \quad \llbracket p_j \rrbracket(p, \tau) = p'}{\langle \text{match } \alpha \ (p_i \Rightarrow c_i)_{i=1, \dots, n}, M, p \cdot P, q \rangle_\nu \rightarrow \langle c_j, M, p' \cdot P, q+1 \rangle_\nu} \\
\text{S-CALL} \quad \frac{\mathbb{F}(f) = \langle \kappa_1, \dots, \kappa_n \rangle \langle \alpha_1, \dots, \alpha_m \rangle (x_1 : s'_1, \dots, x_r : s'_r) = c \quad \langle k_i, P \rangle \Downarrow_{\text{lab } \ell_i} \langle s_i, P \rangle \Downarrow_{\text{sectype } \tau_i} \tau_i \quad \langle e_i, M, P \rangle \Downarrow v_i \quad \langle s'_i, P' \rangle \Downarrow_{\text{sectype } \tau'_i} \tau'_i \quad M = m \cdot M' \quad m' = (\mathbb{I}', |m'|, \nu) \quad P = (p_{\text{var}}, p_{\text{args}}, p_{\text{local}}) \cdot P' \quad p' = (p'_{\text{var}}, p'_{\text{arg}}, p'_{\text{local}}) \quad p'_{\text{var}} = \{ \kappa_i \mapsto \ell_i \mid i = 1, \dots, n \} \cup \{ \alpha_i \mapsto \tau_i \mid i = 1, \dots, m \} \quad p'_{\text{arg}} = \{ x_i \mapsto \tau'_i \mid i = 1, \dots, r \} \quad p'_{\text{local}} = \{ x \mapsto \perp \mid x \in c \} \quad \mathbb{I}' = \{ \delta(x_i) + \text{sp}(m) \mid i = 1, \dots, r \} \cup \{ \text{sp}(m) \} \cup \{ \delta(z) + \text{sp}(m) \mid z \in c \} \quad |m'| = \{ \text{sp}(m) \mapsto (\text{cod}(p_{\text{arg}}), (\text{cod}(p_{\text{local}}), \text{fp}(m)_\nu)) \} \cup \{ \delta(x_i) + \text{sp}(m) \mapsto v_i \mid i = 1, \dots, r \}}{\langle f \langle k_1, \dots, k_n \rangle \langle s_1, \dots, s_m \rangle (e_1, \dots, e_r), M, P, q \rangle_\nu \rightarrow \langle c; \text{epilogue}, m' \cdot M, p' \cdot P, q+1 \rangle_{\nu+1}} \\
\text{S-SEQ-CONT} \quad \frac{c'_1 \neq \text{stop}}{\langle c_1, m, P, q \rangle_\nu \rightarrow \langle c'_1, m', P', q' \rangle_\nu \quad \langle c_1; c_2, m, P, q \rangle_\nu \rightarrow \langle c'_1; c_2, m', P', q' \rangle_\nu} \\
\text{S-SEQ-STOP} \quad \frac{}{\langle c_1, m, P, q \rangle_\nu \rightarrow \langle \text{stop}, m', P', q' \rangle_\nu \quad \langle c_1; c_2, m, P, q \rangle_\nu \rightarrow \langle c_2, m', P', q' \rangle_\nu} \\
\text{S-INST} \quad \frac{}{\langle c, m, P, h, q \rangle_\nu \rightarrow \langle c', m', P', h', q' \rangle_\nu \quad \langle c, m, P, h, q \rangle_\nu \rightarrow \langle c', m', P', h', q' \rangle_\nu}
\end{array}$$

Not entirely good
Isn't it?

Fig. 23: Small-step relation for commands.

$$\begin{array}{c}
\llbracket \text{int}_\kappa \rrbracket(p, \text{int}_\ell) = p[\kappa \mapsto \ell] \quad \frac{\llbracket p_1 \rrbracket(p, \tau_1) = p' \quad \llbracket p_2 \rrbracket(p', \tau_2) = p''}{\llbracket (p_1 @ p_2)_\kappa \rrbracket(p, (\tau_1 @ \tau_2)_\ell) = p''[\kappa \mapsto \ell]} \quad \frac{\llbracket p \rrbracket(p, \tau) = p' \quad p'' = p'[\kappa_1 \mapsto \ell_1, \kappa_2 \mapsto \ell_2]}{\llbracket (\kappa_1 \mapsto p)_{\kappa_2} \rrbracket(p, \ell_1 \mapsto \tau_{\ell_2}) = p''} \\
\\
\frac{\begin{array}{c} |\bar{\tau}| = n \quad |\bar{p}| = m \quad m \leq n \quad p_0 = p \\ \forall i \in \{1, \dots, m-1\} . \llbracket p_i \rrbracket(p_{i-1}, \tau_i) = p_i \\ \llbracket p_m \rrbracket(p_{m-1}, \tau_{m \dots n}) = p' \end{array}}{\llbracket \bar{p} \rrbracket(p, \bar{\tau}) = p'} \\
\\
\llbracket \alpha \rrbracket(p, \tau) = p[\alpha \mapsto \tau]
\end{array}$$

The semantics needs to compute the size of a runtime representation of a time, which is computed using the function $|\cdot|$, that optionally returns an undefined value \perp when invoked on nonsense types $\not\downarrow$.

$$\begin{array}{c}
|\cdot| : \tau \rightarrow \mathbb{N}_\perp \\
|\pi_\ell| = 1 \\
|\tau_1, \dots, \tau_n| = \sum_{i=1}^n |\tau_i| \\
|\not\downarrow| = \perp
\end{array}$$

where $\perp + n = n + \perp = \perp$ for all $n \in \mathbb{N}$.

$$\langle e, m, p \rangle \Downarrow v$$

$$\begin{array}{c}
\begin{array}{cc}
\text{E-NUM} & \text{E-VAR} \\
\frac{}{\langle n, m, p \rangle \Downarrow n} & \frac{m(\delta(x) + \text{fp}(m)) = v}{\langle x, m, p \rangle \Downarrow v}
\end{array}
\quad
\begin{array}{c}
\text{E-BINOP} \\
\frac{\langle e_i, m, p \rangle \Downarrow v_i \quad v_1 \oplus v_2 = v}{\langle e_1 \oplus e_2, m, p \rangle \Downarrow v}
\end{array}
\quad
\begin{array}{c}
\text{E-NUL} \\
\frac{\nu = \nu(m)}{\langle \text{null}, m, p \rangle \Downarrow 0_\nu}
\end{array}
\quad
\begin{array}{c}
\text{E-SIZEOF} \\
\frac{\langle s, p \rangle \Downarrow_{\text{sectype } \tau}}{\langle \text{sizeof } s, m, p \rangle \Downarrow |\tau|}
\end{array} \\
\\
\begin{array}{cc}
\text{E-PACK-TY} & \text{E-PACK-LEV} \\
\frac{\langle s, p \rangle \Downarrow_{\text{sectype } \tau} \quad \langle e, m, p \rangle \Downarrow v}{\langle \text{pack } (s, e) \text{ as } _, m, p \rangle \Downarrow (\tau, v)} & \frac{\langle k, p \rangle \Downarrow_{\text{lab } \ell} \quad \langle e, m, p \rangle \Downarrow v}{\langle \text{pack } (k, e) \text{ as } _, m, p \rangle \Downarrow (\ell, v)}
\end{array}
\quad
\begin{array}{c}
\text{E-UNROLL} \\
\frac{\langle e, m, p \rangle \Downarrow v}{\langle \text{unroll } e, m, p \rangle \Downarrow v}
\end{array}
\quad
\begin{array}{c}
\text{E-ROLL} \\
\frac{\langle e, m, p \rangle \Downarrow v}{\langle \text{roll } e, m, p \rangle \Downarrow v}
\end{array} \\
\\
\begin{array}{c}
\text{E-ADDR} \\
\frac{\nu = \nu(m)}{\langle \&x, m, p \rangle \Downarrow (\delta(x) + \text{fp}(m))_\nu}
\end{array}
\quad
\begin{array}{c}
\text{E-SIZEOF} \\
\frac{\langle s, p \rangle \Downarrow_{\text{sectype } \tau}}{\langle \text{sizeof } s, m, p \rangle \Downarrow |\tau|}
\end{array}
\end{array}$$

$$\langle s, p \rangle \Downarrow_{\text{sectype } \tau}$$

$$\begin{array}{cc}
\text{E-SECTY-SECTY} & \text{E-SECTY-PROD} \\
\frac{\langle t, p \rangle \Downarrow_{\text{type } \pi} \quad \langle k, p \rangle \Downarrow_{\text{lab } \ell}}{\langle tk, p \rangle \Downarrow_{\text{sectype } \pi_\ell}} & \frac{\langle s_i, p \rangle \Downarrow_{\text{sectype } \tau_i} \quad i = 1, \dots, n}{\langle \bar{s}, p \rangle \Downarrow_{\text{sectype } \bar{\tau}}}
\end{array}
\quad
\begin{array}{c}
\text{E-SECTY-VAR} \\
\frac{p(\alpha) = \tau}{\langle \alpha, p \rangle \Downarrow_{\text{sectype } \tau}}
\end{array}$$

$$\langle k, p \rangle \Downarrow_{\text{lab } \ell}$$

$$\begin{array}{cc}
\text{E-LEV-VAR} & \text{E-LEV-JOIN} \\
\frac{p(\kappa) = \ell}{\langle \kappa, p \rangle \Downarrow_{\text{lab } \ell}} & \frac{\langle k_i, p \rangle \Downarrow_{\text{lab } \ell_i}}{\langle k_1 \sqcup k_2, p \rangle \Downarrow_{\text{lab } \ell_1 \sqcup \ell_2}}
\end{array}
\quad
\begin{array}{c}
\text{E-LEV-MEET} \\
\frac{\langle k_i, p \rangle \Downarrow_{\text{lab } \ell_i}}{\langle k_1 \sqcap k_2, p \rangle \Downarrow_{\text{lab } \ell_1 \sqcap \ell_2}}
\end{array}
\quad
\begin{array}{c}
\text{E-LEV-LIT} \\
\langle \ell, p \rangle \Downarrow_{\text{lab } \ell}
\end{array}$$

$$\langle t, p \rangle \Downarrow_{\text{type } \pi}$$

E-TY-INT
 $\langle \text{int}, p \rangle \Downarrow_{\text{type}} \text{int}$

E-TY-PTR
 $\frac{\langle k, p \rangle \Downarrow_{\text{lab}} \ell \quad \langle s, p \rangle \Downarrow_{\text{sectype}} \tau}{\langle k \mapsto s, p \rangle \Downarrow_{\text{type}} \ell \mapsto \tau}$

E-TY-SPTR
 $\frac{\langle s_i, p \rangle \Downarrow_{\text{sectype}} \tau_i \quad i = 1, 2}{\langle s_1 @ s_2, p \rangle \Downarrow_{\text{type}} \tau_1 @ \tau_2}$

E-TY-EX-TY
 $\langle \exists \alpha : \text{type}_k. s, p \rangle \Downarrow_{\text{type}} \exists \alpha : \text{type}. s$

E-TY-EX-LEV
 $\langle \exists \kappa : \text{level}_k. s, p \rangle \Downarrow_{\text{type}} \exists \kappa : \text{level}. s$

E-TY-RBC
 $\langle \mu \alpha : \text{type}_k. s, p \rangle \Downarrow_{\text{type}} \mu \alpha : \text{type}. s$

E-TY-SIZEOF
 $\frac{\langle s, p \rangle \Downarrow_{\text{sectype}} \tau}{\langle \text{size}[s], p \rangle \Downarrow_{\text{type}} \text{size}[\tau]}$

$T_{\text{st}}(pc, fr, k) = (\mu \alpha : \text{type}_k. (\exists \beta : \text{type}_{fr}. (\exists \gamma : \text{type}_{fr}. (\beta \cdot \alpha @ \gamma)_{pc})_{\perp})_{\perp})_{\perp}$

$\Gamma, \Pi, \phi, pc, fr \vdash c$

T-LET
 $\frac{\Gamma, \Pi, \phi \vdash e : r \quad \Pi, \phi \vdash_{\text{sectype}} s : k \quad \phi \vdash r^{pc} <: s \quad fr' = fr \sqcup k \quad \Gamma[x \mapsto s], \Pi, \phi, pc, fr' \vdash c}{\Gamma, \Pi, \phi, pc, fr \vdash \text{let } x : s := e \text{ in } c}$

T-AT
 $\frac{\Pi; \phi \vdash_{\text{lab}} k : pc \quad \Gamma, \Pi, \phi \vdash e : \text{int}_{pc} \quad \phi \vdash pc \sqsubseteq k \quad \Gamma, \Pi, \phi, k, fr \vdash c}{\Gamma, \Pi, \phi, pc, fr \vdash \text{at } k \text{ e } c}$

T-IF
 $\frac{\Gamma, \Pi, \phi \vdash e : \text{int}_{pc} \quad \Gamma, \Pi, \phi, pc, fr \vdash c_i \quad i = 1, 2}{\Gamma, \Pi, \phi, pc, fr \vdash \text{if } e \text{ c}_1 \text{ c}_2}$

T-FP
 $\frac{\Pi; \phi \vdash_{\text{lab}} fr : k \quad \phi \vdash T_{\text{st}}(pc, fr, k)^{pc} <: \Gamma(x)}{\Gamma, \Pi, \phi, pc, fr \vdash x := \text{fp}}$

T-MATCH
 $\frac{\Pi(\alpha) = \text{type}_k \quad \phi \vdash k \sqsubseteq pc \quad \Pi \vdash p_i \rightsquigarrow_k \Pi_i : s_i \quad \Gamma[s_i/\alpha], \Pi_i[s_i/\alpha], \phi, pc, fr \vdash c_i[s_i/\alpha]}{\Gamma, \Pi, \phi, pc, fr \vdash \text{match } \alpha \bar{p} \Rightarrow \bar{c}}$

T-UNPACK-TY
 $\frac{\Gamma, \Pi, \phi \vdash e : (\exists \alpha : \text{type}_{k_1}. r)_{pc} \quad \phi \vdash r^{pc} <: s \quad \Gamma' = \Gamma[x \mapsto s] \quad \Pi' = \Pi[\alpha \mapsto \text{type}_{k_1}] \quad \Pi', \phi \vdash_{\text{sectype}} r : k_2 \quad fr' = fr \sqcup k_1 \sqcup k_2 \quad \Gamma', \Pi', \phi, pc, fr' \vdash c}{\Gamma, \Pi, \phi, pc, fr \vdash \text{let } (\alpha : \text{type}_{k_1}, x : s) := e \text{ in } c}$

T-UNPACK-LEV
 $\frac{\Gamma, \Pi, \phi \vdash e : (\exists \kappa : \text{level}_{k_1}. r)_{pc} \quad \phi \vdash r^{pc} <: s \quad \Gamma' = \Gamma[x \mapsto s] \quad \Pi' = \Pi[\kappa \mapsto \text{level}_{k_1}] \quad \Pi', \phi \vdash_{\text{sectype}} r : k_2 \quad fr' = fr \sqcup k_1 \sqcup k_2 \quad \Gamma', \Pi', \phi, pc, fr' \vdash c}{\Gamma, \Pi, \phi, pc, fr \vdash \text{let } (\kappa : \text{level}_{k_1}, x : s) := e \text{ in } c}$

T-FLOWSTO
 $\frac{\Pi; \phi \vdash_{\text{lab}} k_i : pc \quad \Gamma, \Pi, \phi \wedge k_1 \sqsubseteq k_2, pc, fr \vdash c_1 \quad \Gamma, \Pi, \phi \wedge k_1 \not\sqsubseteq k_2, pc, fr \vdash c_2}{\Gamma, \Pi, \phi, pc, fr \vdash \text{if } (k_1 \sqsubseteq k_2) \text{ c}_1 \text{ c}_2}$

T-INST-C
 $\frac{\Gamma, \Pi, \phi, pc, fr \vdash c}{\Gamma, \Pi, \phi, pc, fr \vdash c}$

T-WHILE
 $\frac{\Gamma, \Pi, \phi \vdash e : \text{int}_{pc} \quad \Gamma, \Pi, \phi, pc, fr \vdash c}{\Gamma, \Pi, \phi, pc, fr \vdash \text{while } e \text{ c}}$

T-SEQ
 $\frac{\Gamma, \Pi, \phi, pc, fr \vdash c_i \quad i = 1, 2}{\Gamma, \Pi, \phi, pc, fr \vdash c_1; c_2}$

T-ASSIGN
 $\frac{\Gamma, \Pi, \phi \vdash e : s \quad \phi \vdash s^{pc} <: \Gamma(x)}{\Gamma, \Pi, \phi, pc, fr \vdash x := e}$

T-CALL
 $\frac{\mathbb{F}(f) = \langle \kappa_1 : k_1^1, \dots, \kappa_n : k_n^1 \rangle \langle \alpha_1 : k_1^2, \dots, \alpha_m : k_m^2 \rangle (x_1 : s_1, \dots, x_r : s_r) \rightarrow_{k_{pc}}^{k_{fr}} 1 \quad \Pi; \phi \vdash_{\text{lab}} k_i : k_i^1[k_{i-1}/\kappa_{i-1}, \dots, k_1/\kappa_1] \quad \Pi, \phi \vdash_{\text{sectype}} s_i : k_i^2[k_n/\kappa_n, \dots, k_1/\kappa_1][s_{i-1}/\alpha_{i-1}, \dots, s_1/\alpha_1] \quad \Gamma, \Pi, \phi \vdash e_i : s_i[k_n/\kappa_n, \dots, k_1/\kappa_1][s_m/\alpha_m, \dots, s_1/\alpha_1] \quad \phi \vdash k_{pc}[k_n/\kappa_n, \dots, k_1/\kappa_1] = pc \quad \phi \vdash fr \sqsubseteq k_{fr}[k_n/\kappa_n, \dots, k_1/\kappa_1]}{\Gamma, \Pi, \phi, pc, fr \vdash f\langle k_1, \dots, k_n \rangle \langle s_1, \dots, s_m \rangle (e_1, \dots, e_r)}$

Relation $\Pi \vdash p \rightsquigarrow_k \Pi' : s$ specifies that a matchee can safely be assigned type s if the runtime value matches the pattern p assuming environment Π is updated to Π' . Finally, the label k represents an upper bound on the information that influences the

type of s .

$$\boxed{\Pi \vdash p \rightsquigarrow_k \Pi' : s}$$

$$\Pi \vdash \text{int}_\kappa \rightsquigarrow_k \Pi[\kappa \mapsto \text{level}_k] : \text{int}_\kappa \quad \Pi \vdash \alpha \rightsquigarrow_k \Pi[\alpha \mapsto \text{type}_k] : \alpha$$

$$\frac{\Pi \vdash p_1 \rightsquigarrow_k \Pi_1 : s_1 \quad \Pi_1 \vdash p_2 \rightsquigarrow_k \Pi_2 : s_2}{\Pi \vdash (p_1 @ p_2)_\kappa \rightsquigarrow_k \Pi''[\kappa \mapsto \text{level}_k] : (s_1 @ s_2)_\kappa}$$

$$\frac{\Pi \vdash p \rightsquigarrow_k \Pi' : s}{\Pi \vdash (\kappa_1 \mapsto p)_{\kappa_2} \rightsquigarrow_k \Pi'[\kappa_1 \mapsto \text{level}_k, \kappa_2 \mapsto \text{level}_k] : (\kappa_1 \mapsto s)_{\kappa_2}}$$

$$\frac{\Pi_0 = \Pi \quad \Pi_{i-1} \vdash p_i \rightsquigarrow_k \Pi_i : s_i \quad i = 1, \dots, n}{\Pi \vdash p_1, \dots, p_n \rightsquigarrow_k \Pi_n : s_1, \dots, s_n}$$

Relation $\Gamma, \Pi, \phi \vdash e : s$ specifies that expression e has type s in the typing environment Γ and assuming the constraints ϕ .

$$\boxed{\Gamma, \Pi, \phi \vdash e : s}$$

T-NUM

$$\frac{}{\Gamma, \Pi, \phi \vdash n : \text{int}_\perp}$$

T-VAR

$$\frac{}{\Gamma, \Pi, \phi \vdash x : \Gamma(x)}$$

T-PACK-TY

$$\frac{\Pi, \phi \vdash_{\text{sectype}} s_2' \quad \Gamma, \Pi, \phi \vdash e : s_2[s_1/x] \quad \Pi, \phi \vdash_{\text{sectype}} s_1 : k_1 \quad t = \exists x : \text{type}_{k_1} s_2}{\Gamma, \Pi, \phi \vdash \text{pack}(s_1, e) \text{ as } t : t_\perp}$$

T-SIZEOF

$$\frac{\Pi, \phi \vdash_{\text{sectype}} s : k}{\Gamma, \Pi, \phi \vdash \text{sizeof } s : \text{size}[s]_k}$$

T-BINOP

$$\frac{\Gamma, \Pi, \phi \vdash e_i : s_i \quad s_1 \llbracket \oplus \rrbracket s_2 \rightarrow s}{\Gamma, \Pi, \phi \vdash e_1 \oplus e_2 : s}$$

T-UNROLL

$$\frac{\Gamma, \Pi, \phi \vdash e : (\mu \alpha : \text{type}_{k_1} s)_{k_2}}{\Gamma, \Pi, \phi \vdash \text{unroll } e : s[(\mu \alpha : \text{type}_{k_1} s)_{k_2} / \alpha]_{k_2}}$$

T-ROLL

$$\frac{\Gamma, \Pi, \phi \vdash e : s[(\mu \alpha : \text{type}_{k_1} s)_{k_2} / \alpha]_{k_3}}{\Gamma, \Pi, \phi \vdash \text{roll } e : (\mu \alpha : \text{type}_{k_1} s)_{k_2 \sqcup k_3}}$$

T-NUL-HEAP

$$\frac{\Pi, \phi \vdash_{\text{sectype}} s \quad \Pi; \phi \vdash_{\text{lab}} k}{\Gamma, \Pi, \phi \vdash \text{null} : (k \mapsto s)_\perp}$$

T-NUL-STACK

$$\frac{\Pi, \phi \vdash_{\text{sectype}} s_i \quad i = 1, 2}{\Gamma, \Pi, \phi \vdash \text{null} : (s_1 @ s_2)_\perp}$$

T-PACK-LEV

$$\frac{\Pi, \phi \vdash_{\text{sectype}} s : _ \quad \Gamma, \Pi, \phi \vdash e : s[k/\kappa] \quad \Pi; \phi \vdash_{\text{lab}} k : k' \quad t = \exists \kappa : \text{level}_{k'} s}{\Gamma, \Pi, \phi \vdash \text{pack}(k, e) \text{ as } t : t_\perp}$$

T-ADDR OF

$$\frac{\Gamma(x) = s}{\Gamma, \Pi, \phi \vdash \&x : (@ s)_\perp}$$

T-SUB

$$\frac{\Gamma, \Pi, \phi \vdash e : s_1 \quad \phi \vdash s_1 <: s_2}{\Gamma, \Pi, \phi \vdash e : s_2}$$

T-CONV

$$\frac{\Gamma, \Pi, \phi \vdash e : \text{size}[s]_k}{\Gamma, \Pi, \phi \vdash e : \text{int}_k}$$

$$\boxed{\Pi, \phi \vdash_{\text{sectype}} s : k}$$

T-SEC-TYPE

$$\frac{\Pi, \phi \vdash_{\text{type}} t : k_1 \quad \Pi; \phi \vdash_{\text{lab}} k : k_2}{\Pi, \phi \vdash_{\text{sectype}} t_k : k_1 \sqcup k_2}$$

T-SEC-VAR

$$\frac{\Pi(\alpha) = \text{type}_{k_1} \quad \Pi; \phi \vdash_{\text{lab}} k : k_2}{\Pi, \phi \vdash_{\text{sectype}} \alpha^k : k_1 \sqcup k_2}$$

T-SEC-PROD

$$\frac{\Pi, \phi \vdash_{\text{sectype}} s_i : k_i \quad i = 1, \dots, n}{\Pi, \phi \vdash_{\text{sectype}} s_1, \dots, s_n : \sqcup_{i=1, \dots, n} k_i}$$

T-SEC-SUB

$$\frac{\Pi, \phi \vdash_{\text{sectype}} s : k_1 \quad \phi \vdash k_1 \sqsubseteq k_2}{\Pi, \phi \vdash_{\text{sectype}} s : k_2}$$

$$\boxed{\Pi, \phi \vdash_{\text{type}} t : k}$$

$$\begin{array}{c}
\text{T-BASE-INT} \quad \checkmark \\
\frac{}{\Pi, \phi \vdash_{\text{type}} \text{int} : \perp}
\end{array}
\quad
\begin{array}{c}
\text{T-BASE-SPTR} \quad \checkmark \\
\frac{\Pi, \phi \vdash_{\text{sectype}} s_i : k_i}{\Pi, \phi \vdash_{\text{type}} s_1 @ s_2 : k_1 \sqcup k_2}
\end{array}
\quad
\begin{array}{c}
\text{T-BASE-PTR} \\
\frac{\Pi; \phi \vdash_{\text{lab}} k : k_1 \quad \Pi, \phi \vdash_{\text{sectype}} s : k_2}{\Pi, \phi \vdash_{\text{type}} k \mapsto s : k_1 \sqcup k_2}
\end{array}$$

$$\begin{array}{c}
\text{T-BASE-EX-TY} \\
\frac{\Pi; \phi \vdash_{\text{lab}} k : k_1 \quad \Pi[\alpha \mapsto \text{type}_k], \phi \vdash_{\text{sectype}} s : k_2}{\Pi, \phi \vdash_{\text{type}} \exists \alpha : \text{type}_k. s : k_1 \sqcup k_2}
\end{array}
\quad
\begin{array}{c}
\text{T-BASE-EX-LEV} \quad \checkmark \\
\frac{\Pi; \phi \vdash_{\text{lab}} k : k_1 \quad \Pi[\kappa \mapsto \text{level}_k], \phi \vdash_{\text{sectype}} s : k_2}{\Pi, \phi \vdash_{\text{type}} \exists \kappa : \text{type}_k. s : k_1 \sqcup k_2}$$

$$\begin{array}{c}
\text{T-BASE-MU} \quad \checkmark \\
\frac{\phi \vdash k_1 \sqsubseteq k' \quad \Pi[\alpha \mapsto \text{type}_k]; \phi \vdash_{\text{lab}} s : k' \quad \Pi; \phi \vdash_{\text{lab}} k_1 : k'}{\Pi, \phi \vdash_{\text{type}} (\mu \alpha : \text{type}_k. s) : k'}
\end{array}$$

$$\begin{array}{c}
\text{T-BASE-SIZE} \quad \checkmark \\
\frac{\Pi, \phi \vdash_{\text{sectype}} s : k}{\Pi, \phi \vdash_{\text{type}} \text{size}[s] : k}
\end{array}
\quad
\begin{array}{c}
\text{T-BASE-SUB} \quad \checkmark \\
\frac{\Pi, \phi \vdash_{\text{type}} s : k_1 \quad \phi \vdash k_1 \sqsubseteq k_2}{\Pi, \phi \vdash_{\text{type}} s : k_2}$$

$$\begin{array}{c}
\text{T-LAB-LIT} \\
\Pi; \phi \vdash_{\text{lab}} \ell : \perp
\end{array}
\quad
\begin{array}{c}
\text{T-LAB-VAR} \\
\frac{\Pi(\kappa) = \text{level}_k}{\Pi; \phi \vdash_{\text{lab}} \kappa : k}$$

$$\begin{array}{c}
\text{T-LAB-JOIN} \\
\frac{\Pi; \phi \vdash_{\text{lab}} k_i : k'_i \quad i = 1, 2}{\Pi; \phi \vdash_{\text{lab}} k_1 \sqcup k_2 : k'_1 \sqcup k'_2}$$

$$\begin{array}{c}
\text{T-LAB-MEET} \\
\frac{\Pi; \phi \vdash_{\text{lab}} k_i : k'_i \quad i = 1, 2}{\Pi; \phi \vdash_{\text{lab}} k_1 \sqcap k_2 : k'_1 \sqcap k'_2}$$

$$\begin{array}{c}
\text{T-LAB-SUB} \\
\frac{\phi \vdash k_1 \sqsubseteq k_2 \quad \Pi; \phi \vdash_{\text{lab}} k : k_1}{\Pi; \phi \vdash_{\text{lab}} k : k_2}
\end{array}$$

Where does ℓ , can be any and where is ℓ itself used?

Relation $s_1 \llbracket s_2 \rrbracket \oplus \rightarrow s$ computes the type s of the result of evaluating a binary expression \oplus on two expressions of type s_1 and s_2 respectively.

$$\begin{array}{c}
\boxed{s_1 \llbracket s_2 \rrbracket \oplus \rightarrow s} \\
\\
\text{int}_{k_1} \llbracket \oplus \rrbracket \text{int}_{k_2} \rightarrow \text{int}_{(k_1 \sqcup k_2)} \\
(s_1 @ s \cdot s_2)_{k_1} \llbracket + \rrbracket \text{size}[s]_{k_2} \rightarrow (s_1 \cdot s @ s_2)_{(k_1 \sqcup k_2)} \\
(s_1 \cdot s @ s_2)_{k_1} \llbracket - \rrbracket \text{size}[s]_{k_2} \rightarrow (s_1 @ s \cdot s_2)_{(k_1 \sqcup k_2)} \\
(k \mapsto s)_{k_1} \llbracket + \rrbracket \text{int}_{k_2} \rightarrow (k \mapsto s)_{(k_1 \sqcup k_2)} \\
(k \mapsto s)_{k_1} \llbracket - \rrbracket \text{int}_{k_2} \rightarrow (k \mapsto s)_{(k_1 \sqcup k_2)} \\
\\
\boxed{P \models \phi} \\
\frac{\langle k_i, P \rangle \Downarrow_{\text{lab}} \ell_i \quad i = 1, 2 \quad \ell_1 \sqsubseteq \ell_2}{P \models k_1 \sqsubseteq k_2} \quad
\frac{\langle k_i, P \rangle \Downarrow_{\text{lab}} \ell_i \quad i = 1, 2 \quad \ell_1 \not\sqsubseteq \ell_2}{P \models k_1 \not\sqsubseteq k_2} \quad
\frac{P \models \phi_i \quad i = 1, 2}{P \models \phi_1 \wedge \phi_2} \\
\\
\boxed{\Gamma, \Pi, \phi \models \langle c, M, P, h, q \rangle_\nu} \\
\Gamma, \Pi, \phi \models \langle c, M, P, h, q \rangle_\nu \iff \Gamma, \Pi, \phi \models (M, P, h) \\
\frac{}{\Gamma, \Pi, \phi \models (\varepsilon, \varepsilon, h)} \quad
\frac{\Gamma, \Pi, \phi \models (m, p, h) \quad \Gamma, \Pi, \phi \models (M, P, h)}{\Gamma, \Pi, \phi \models (m \cdot M, p \cdot P, h)}
\end{array}$$

$$\begin{array}{l}
\Gamma, \Pi, \phi \models (m, p, h) \iff \\
(\forall x. \Gamma(x) = s \wedge m(\delta(x) + \text{fp}(m)) = v \wedge \langle s, p \rangle \Downarrow_{\text{sectype}} \tau \implies \Gamma, \Pi, \phi, M, P, h \models v : \tau) \quad \wedge \\
(\forall \alpha. \Pi(\alpha) = \text{type}_k \wedge \alpha \in \text{dom}(p) \implies \exists \tau. p(\alpha) = \tau) \quad \wedge \\
(\forall \kappa. \Pi(\kappa) = \text{level}_k \wedge \kappa \in \text{dom}(p) \implies \exists \ell. p(\kappa) = \ell) \\
\Gamma, \Pi, \phi, M, P, h \models v : \tau \iff \forall z. \Gamma, \Pi, \phi, M, P, h \models^z v : \tau
\end{array}$$

$$\boxed{\Gamma, \Pi, \phi, M, P, h \models v : \tau}$$

$$\begin{array}{c}
\overline{\Gamma, \Pi, \phi, M, P, h \models^0 v : \tau} \quad \overline{\Gamma, \Pi, \phi, M, P, h \models^z n : \text{int}_\ell} \quad \overline{\Gamma, \Pi, \phi, M, P, h \models^z n : \text{size}[\tau]_\ell} \\
\\
\frac{M = M_1 \cdot m \cdot M_2 \quad m(a) = v \implies \Gamma, \Pi, \phi, M, P, h \models^z v : \tau_2 \quad \Gamma, \Pi, \phi, M, P, h \models^z (a - |\tau_1|) : \tau_1}{\Gamma, \Pi, \phi, M, P, h \models^z a_\nu : (\tau_1 @ \tau_2)_\ell} \quad \frac{\Gamma, \Pi, \phi, P, h \models^z a_\nu : (\ell_p \mapsto \tau)_\ell}{\Gamma, \Pi, \phi, M, P, h \models^z a_\nu : (\ell_p \mapsto \tau)_\ell} \\
\\
\frac{\langle s[\ell/\kappa], P \rangle \Downarrow_{\text{sectype}} \tau \quad \Gamma, \Pi, \phi, M, P, h \models^z v : \tau^{\ell'}}{\Gamma, \Pi, \phi, M, P, h \models (\ell, v) : (\exists \kappa : \text{level. } s)_{\ell'}[z]} \quad \frac{\langle s[\tau/\alpha], P \rangle \Downarrow_{\text{sectype}} \tau' \quad \Gamma, \Pi, \phi, M, P, h \models^z v : \tau'^\ell}{\Gamma, \Pi, \phi, M, P, h \models^z (\tau, v) : (\exists \alpha : \text{type. } s)_\ell} \\
\\
\frac{\langle s[(\mu \alpha : \text{type. } s)_\ell/\alpha], P \rangle \Downarrow_{\text{sectype}} \tau \quad \Gamma, \Pi, \phi, M, P, h \models^{z-1} v : \tau}{\Gamma, \Pi, \phi, M, P, h \models^z v : (\mu \alpha : \text{type. } s)_\ell}
\end{array}$$

$$\boxed{\Gamma, \Pi, \phi, M, P, h \models^z a : \bar{\tau}}$$

$$\frac{M = M_1 \cdot m \cdot M_2 \quad m(a) = v \implies \Gamma, \Pi, \phi, M, P, h \models^z a : \tau_1 \quad \Gamma, \Pi, \phi, M, P, h \models^z a + |\tau_1| : \tau_2, \dots, \tau_n}{\Gamma, \Pi, \phi, M, P, h \models^z a : \tau_1, \tau_2, \dots, \tau_n} \quad \overline{\Gamma, \Pi, \phi, M, P, h \models^z a : \varepsilon}$$

In this section we define the attacker model that we consider in this work.

A. Events and event semantics

We define a semantics augmented with events, given by the following grammar:

$$\begin{aligned}
ev ::= & \varepsilon \mid \text{asgn}(x \leftarrow v, q) \mid \text{rd}(x \leftarrow v, q) \\
& \mid \text{unp}(\ell, x : \tau \leftarrow v, q) \mid \text{let}(x : \tau \leftarrow v, q) \mid ev
\end{aligned}$$

B. \mathcal{A} -observability and \mathcal{A} -equivalences

$$\boxed{p_1, p_2 \vdash v_1 =_{\mathcal{A}}^z v_2 : \tau_1 \times \tau_2}$$

EQ-VAL-0

$$\frac{}{p_1, p_2 \vdash v_1 =_{\mathcal{A}}^0 v_2 : \tau_1 \times \tau_2}$$

EQ-INT-LOW

$$\frac{\ell \sqsubseteq \mathcal{A}}{p_1, p_2 \vdash n =_{\mathcal{A}}^z n : \text{int}_{\ell} \times \text{int}_{\ell}}$$

EQ-INT-HIGH

$$\frac{\ell_i \not\sqsubseteq \mathcal{A} \quad i = 1, 2}{p_1, p_2 \vdash n_1 =_{\mathcal{A}}^z n_2 : \text{int}_{\ell_1} \times \text{int}_{\ell_2}}$$

EQ-SPTR-LOW

$$\frac{\ell \sqsubseteq \mathcal{A}}{p_1, p_2 \vdash a_{\nu} =_{\mathcal{A}}^z a_{\nu} : (\tau_1 @ \tau_2)_{\ell} \times (\tau_1 @ \tau_2)_{\ell}}$$

EQ-SPTR-HIGH

$$\frac{\ell_i \not\sqsubseteq \mathcal{A}}{p_1, p_2 \vdash a_{\nu} =_{\mathcal{A}}^z a_{\nu} : (\tau_1^1 @ \tau_2^1)_{\ell_1} \times (\tau_1^2 @ \tau_2^2)_{\ell_2}}$$

EQ-PTR-LOW

$$\frac{\ell' \sqsubseteq \mathcal{A}}{p_1, p_2 \vdash a_{\nu} =_{\mathcal{A}}^z a_{\nu} : (\ell \mapsto \tau)_{\ell'} \times (\ell \mapsto \tau)_{\ell'}}$$

EQ-PTR-HIGH

$$\frac{\ell'_i \not\sqsubseteq \mathcal{A}}{p_1, p_2 \vdash a_{\nu} =_{\mathcal{A}}^z a_{\nu} : (\ell_1 \mapsto \tau_1)_{\ell'_1} \times (\ell_2 \mapsto \tau_2)_{\ell'_2}}$$

EQ-EX-TY-LOW

$$\frac{\langle s'_i[\tau_i/\alpha], p_i \rangle \Downarrow_{\text{sectype}} \tau_i'' \quad p_1, p_2 \vdash v_1 =_{\mathcal{A}}^z v_2 : \tau_1' \times \tau_2' \quad \ell \sqsubseteq \mathcal{A} \quad \tau_i' = (\exists \alpha : \text{type}_{\ell'_i}. s_i)_{\ell} \quad i = 1, 2 \quad p_1, p_2 \vdash \tau_1 =_{\mathcal{A}}^z \tau_2 : \ell'_1 \times \ell'_2}{p_1, p_2 \vdash (\tau_1, v_1) =_{\mathcal{A}}^z (\tau_2, v_2) : \tau_1' \times \tau_2'}$$

EQ-EX-TY-HIGH

$$\frac{\ell_i \not\sqsubseteq \mathcal{A} \quad i = 1, 2 \quad \tau_i' = (\exists \alpha : \text{type}_{\ell'_i}. s_i)_{\ell_i} \quad i = 1, 2 \quad p_1, p_2 \vdash \tau_1 =_{\mathcal{A}}^z \tau_2 : \ell'_1 \times \ell'_2}{p_1, p_2 \vdash (\tau_1, v_1) =_{\mathcal{A}}^z (\tau_2, v_2) : \tau_1' \times \tau_2'}$$

EQ-EX-LEV-LOW

$$\frac{\langle s'_i[\ell_i/\kappa], p_i \rangle \Downarrow_{\text{sectype}} \tau_i'' \quad p_1, p_2 \vdash v_1 =_{\mathcal{A}}^z v_2 : \tau_1' \times \tau_2' \quad \ell \sqsubseteq \mathcal{A} \quad \tau_i' = (\exists \kappa : \text{level}_{\ell'_i}. s_i)_{\ell} \quad i = 1, 2 \quad p_1, p_2 \vdash \ell_1 =_{\mathcal{A}}^z \ell_2 : \ell'_1 \times \ell'_2}{p_1, p_2 \vdash (\ell_1, v_1) =_{\mathcal{A}}^z (\ell_2, v_2) : \tau_1' \times \tau_2'}$$

EQ-EX-LEV-HIGH

$$\frac{\ell'_i \not\sqsubseteq \mathcal{A} \quad i = 1, 2 \quad \tau_i' = (\exists \kappa : \text{level}_{\ell'_i}. s_i)_{\ell'_i} \quad i = 1, 2 \quad p_1, p_2 \vdash \ell_1 =_{\mathcal{A}}^z \ell_2 : \ell'_1 \times \ell'_2}{p_1, p_2 \vdash (\ell_1, v_1) =_{\mathcal{A}}^z (\ell_2, v_2) : \tau_1' \times \tau_2'}$$

EQ-SIZE-LOW

$$p_1, p_2 \vdash n =_{\mathcal{A}} n : \text{size}[\tau]_{\ell} \times \text{size}[\tau]_{\ell}$$

EQ-SIZE-HIGH

$$p_1, p_2 \vdash n_1 =_{\mathcal{A}} n_2 : \text{size}[\tau]_{\ell_1} \times \text{size}[\tau]_{\ell_2}$$

EQ-REC

$$\frac{p_1, p_2 \vdash \ell_1 =_{\mathcal{A}} \ell_2 : \ell_1 \times \ell_2 \quad p_1, p_2 \vdash \ell'_1 =_{\mathcal{A}} \ell'_2 : \ell'_1 \times \ell'_2 \quad \langle s_i[(\mu \alpha : \text{type}_{\ell_i}. s_i)/\alpha], p_i \rangle \Downarrow_{\text{sectype}} \tau_i \quad p_1, p_2 \vdash v_1 =_{\mathcal{A}}^{z-1} v_2 : \tau_1 \times \tau_2}{p_1, p_2 \vdash v_1 =_{\mathcal{A}}^z v_2 : (\mu \alpha : \text{type}_{\ell_1}. s_1)_{\ell'_1} \times (\mu \alpha : \text{type}_{\ell_2}. s_2)_{\ell'_2}}$$

$$\boxed{\tau_1 =_{\mathcal{A}} \tau_2 : \ell_1 \times \ell_2}$$

$$\frac{\ell' \sqsubseteq \mathcal{A}}{p_1, p_2 \vdash \pi_{\ell} =_{\mathcal{A}} \pi_{\ell} : \ell' \times \ell'}$$

$$\frac{\ell \sqsubseteq \mathcal{A}}{p_1, p_2 \vdash \bar{\tau} =_{\mathcal{A}} \bar{\tau} : \ell \times \ell}$$

$$\frac{\ell \sqsubseteq \mathcal{A}}{p_1, p_2 \vdash \underline{\tau} =_{\mathcal{A}} \underline{\tau} : \ell \times \ell}$$

$$\frac{\ell_i \not\sqsubseteq \mathcal{A} \quad i = 1, 2}{p_1, p_2 \vdash \tau_1 =_{\mathcal{A}} \tau_2 : \ell_1 \times \ell_2}$$

$$\boxed{\ell_1 =_{\mathcal{A}} \ell_2 : \ell'_1 \times \ell'_2}$$

$$\frac{\ell' \sqsubseteq \mathcal{A}}{\ell =_{\mathcal{A}} \ell : \ell' \times \ell'}$$

$$\frac{\ell'_i \not\sqsubseteq \mathcal{A} \quad i = 1, 2}{\ell_1 =_{\mathcal{A}} \ell_2 : \ell'_1 \times \ell'_2}$$

$$\boxed{\Gamma \mid p_1, p_2 \vdash \text{ev}_1 =_{\mathcal{A}} \text{ev}_2}$$

$$\frac{\langle \Gamma(x), p_i \rangle \Downarrow_{\text{sectype}} \tau_i \quad i = 1, 2 \quad p_1, p_2 \vdash v_1 =_{\mathcal{A}} v_2 : \tau_1 \times \tau_2}{\Gamma \mid p_1, p_2 \vdash \text{asgn}(x \leftarrow v_1, q) =_{\mathcal{A}} \text{asgn}(x \leftarrow v_2, q)}$$

$$\frac{\langle \Gamma(x), p_i \rangle \Downarrow_{\text{sectype}} \tau_i \quad i = 1, 2 \quad p_1, p_2 \vdash v_1 =_{\mathcal{A}} v_2 : \tau_1 \times \tau_2}{\Gamma \mid p_1, p_2 \vdash \text{rd}(x \leftarrow v_1, q) =_{\mathcal{A}} \text{rd}(x \leftarrow v_2, q)}$$

$$\frac{p_1, p_2 \vdash \ell_1 =_{\mathcal{A}} \ell_2 : \ell_1 \times \ell_2 \quad p_1, p_2 \vdash v_1 =_{\mathcal{A}} v_2 : \tau_1 \times \tau_2}{\Gamma \mid p_1, p_2 \vdash \text{unp}(\ell_1, y : \tau_1 \leftarrow v_1, q) =_{\mathcal{A}} \text{unp}(\ell_2, y : \tau_2 \leftarrow v_2, q)}$$

$$\frac{p_1, p_2 \vdash v_1 =_{\mathcal{A}} v_2 : \tau_1 \times \tau_2}{\Gamma \mid p_1, p_2 \vdash \text{let}(x : \tau_1 \leftarrow v_1, q) =_{\mathcal{A}} \text{let}(x : \tau_2 \leftarrow v_2, q)}$$

$$\frac{\Gamma \mid p_1, p_2 \vdash \text{ev}_1 =_{\mathcal{A}} \text{ev}_2}{\Gamma \mid p_1, p_2 \vdash \text{ev}_1 =_{\mathcal{A}} \text{ev}_2}$$

$$\boxed{\tau \sqsubseteq \mathcal{A}}$$

$$\pi_\ell \sqsubseteq \mathcal{A} \iff \ell \sqsubseteq \mathcal{A}$$

$$\boxed{\Gamma, p \vdash ev \sqsubseteq \mathcal{A}}$$

$\frac{\text{EV-OBS-ASGN} \quad \langle \Gamma(x), P \rangle \Downarrow_{\text{sectype}} \tau \quad \tau \sqsubseteq \mathcal{A}}{\Gamma, p \vdash \text{asgn}(x \leftarrow v, q) \sqsubseteq \mathcal{A}}$	$\frac{\text{EV-OBS-RD} \quad \langle \Gamma(x), P \rangle \Downarrow_{\text{sectype}} \tau \quad \tau \sqsubseteq \mathcal{A}}{\Gamma, p \vdash \text{rd}(x \leftarrow v, q) \sqsubseteq \mathcal{A}}$	$\frac{\text{EV-OBS-UNP-1} \quad \ell \sqsubseteq \mathcal{A}}{\Gamma, p \vdash \text{unp}(\ell, y : \tau \leftarrow v, q) \sqsubseteq \mathcal{A}}$
$\frac{\text{EV-OBS-UNP-2} \quad \tau \sqsubseteq \mathcal{A}}{\Gamma, p \vdash \text{unp}(\ell, y : \tau \leftarrow v, q) \sqsubseteq \mathcal{A}}$	$\frac{\text{EV-OBS-DECL} \quad \tau \sqsubseteq \mathcal{A}}{\Gamma, p \vdash \text{let}(x : \tau \leftarrow v, q) \sqsubseteq \mathcal{A}}$	$\frac{\text{EV-OBS-INST} \quad \Gamma, P \vdash ev \sqsubseteq \mathcal{A}}{\Gamma, P \vdash ev \sqsubseteq \mathcal{A}}$

Given a trace t we write $[t]_{\mathcal{A}}$ for the trace containing only low events. It is defined inductively as follows:

$$[\varepsilon]_{\mathcal{A}} = \varepsilon$$

$$[(ev, \Gamma, p) \cdot t]_{\mathcal{A}} = \begin{cases} (ev, \Gamma, p) \cdot [t]_{\mathcal{A}} & \Gamma, p \vdash ev \sqsubseteq \mathcal{A} \\ [t]_{\mathcal{A}} & \text{otherwise} \end{cases}$$

Given two traces t_1, t_2 we say they are \mathcal{A} -equivalent, written $t_1 =_{\mathcal{A}} t_2$, when $([t_1]_{\mathcal{A}})_i =_{\mathcal{A}} ([t_2]_{\mathcal{A}})_i$ for $i = 1, \dots, n$ where $n = |[t_1]_{\mathcal{A}}| = |[t_2]_{\mathcal{A}}|$.

We write $\langle c, m, P, h, q \rangle_{\nu} \xrightarrow{t}_{\mathcal{A}} \langle c', m', P', q' \rangle_{\nu'}$ when $\langle c, m, P, h, q \rangle_{\nu} \xrightarrow{t'} \langle c', m', P', q' \rangle_{\nu'}$ and $t =_{\mathcal{A}} t'$.

Finally, we define a notion of \mathcal{A} -equivalent stack frames:

$$\frac{\forall x, i. \quad \langle \Gamma(x), p_i \rangle \Downarrow_{\text{sectype}}^{\ell_i} \tau_i \wedge m_i(\delta(x) + \text{fp}(m_i)) = v_i \implies p_1, p_2 \vdash v_1 =_{\mathcal{A}} v_2 : \tau_1^{\ell_1} \times \tau_2^{\ell_2}}{\Gamma \vdash (p_1, m_1) =_{\mathcal{A}} (p_2, m_2)}$$

$$\frac{pc \sqsubseteq \mathcal{A} \quad \Gamma \vdash (p_1, m_1) =_{\mathcal{A}} (p_2, m_2) \quad \Gamma \vdash (P_1, M_1)_{\overline{pc_1}} =_{\mathcal{A}} (P_2, M_2)_{\overline{pc_2}}}{\Gamma \vdash (p_1 \cdot P_1, m_1 \cdot M_1)_{pc \cdot \overline{pc_1}} =_{\mathcal{A}} (p_2 \cdot P_2, m_2 \cdot M_2)_{pc \cdot \overline{pc_2}}}$$

$$\frac{pc' \not\sqsubseteq \mathcal{A} \quad \Gamma \vdash (P_1, M_1)_{\overline{pc_1}} =_{\mathcal{A}} (P_2, M_2)_{\overline{pc_2}}}{\Gamma \vdash (P_1, M_1)_{\overline{pc_1}} =_{\mathcal{A}} (p_2 \cdot P_2, m_2 \cdot M_2)_{pc' \cdot \overline{pc_2}}} \quad \frac{pc' \not\sqsubseteq \mathcal{A} \quad \Gamma \vdash (P_1, M_1)_{\overline{pc_1}} =_{\mathcal{A}} (P_2, M_2)_{\overline{pc_2}}}{\Gamma \vdash (p_1 \cdot P_1, m_1 \cdot M_1)_{pc' \cdot \overline{pc_1}} =_{\mathcal{A}} (P_2, M_2)_{\overline{pc_2}}} \quad \Gamma \vdash (\varepsilon, \varepsilon)_{\varepsilon} =_{\mathcal{A}} (\varepsilon, \varepsilon)_{\varepsilon}$$

C. Attacker's knowledge

We write $\langle c, m, P, h, q \rangle_{\nu} \rightarrow^*$ when there exists a configuration $\langle \text{stop}, m', P', h', q' \rangle_{\nu'}$ such that $\langle c, m, P, h, q \rangle_{\nu} \rightarrow^* \langle \text{stop}, m', P', h', q' \rangle_{\nu'}$. We use similar notation for the event semantics.

Given a trace t , the attacker's knowledge $k_{\mathcal{A}}(c, M)Pt$ is the set of stacks such that the execution of c produces an \mathcal{A} -equivalent trace:

$$k_{\mathcal{A}}(c, t) = \left\{ (M, P, \overline{pc}) \mid \langle c, M, P \mid \overline{pc} \rangle \xrightarrow{t}_{\mathcal{A}}^* \right\}$$

Note that a larger attacker knowledge set correspond to an attacker obtaining less information. Smaller sets correspond to a more precise knowledge. To define TINI we write the set of terminating executions as

$$k_{\mathcal{T}}^{\downarrow}(c, M_1, P_1, \overline{pc_1}) = \left\{ (M_2, P_2, \overline{pc_2}) \mid \Gamma \vdash (P_1, M_1)_{\overline{pc_1}} =_{\mathcal{A}} (P_2, M_2)_{\overline{pc_2}} \wedge \langle c, M_2, P_2 \mid \overline{pc_2} \rangle \rightarrow^* \right\}$$

Using attacker knowledge and the set of terminating stack frames, we can define the noninterference policy.

Definition 1 (Termination-insensitive noninterference). *A program c satisfies termination-insensitive interference wrt. typing environment Γ if, for all M and P it holds that $\langle c, M, P \mid \overline{pc} \rangle \xrightarrow{t}_{\mathcal{A}}^*$ implies $k_{\mathcal{A}}(c, t) \supseteq k_{\mathcal{T}}^{\downarrow}(c, M, P, \overline{pc})$.*

In this section we prove the soundness of the type system: That well-typed programs satisfies noninterference:

Theorem 2 (Soundness). *If $\Gamma \vdash c$ then c satisfies Definition 1.*

Ptr-to-self

```
let x: ptr @α := roll (null)
in x := roll (&x)
```

```
let y: ptr @α := pack (null) (?)
while ~ do
```

~~let~~ match α with

```
1 b → let x: b := ~
in if i = 5 then
  y := pack (&x, b)
  clear 52, p
```

```
in let (x, b) = Unpack (y)
let x: b = *x_ptr
```

~~either~~ either
This is not
type-safe
or, it's blowing up
the stack.

Possible solution
is to perform
versioning on
nested contexts

```
let x: α := null null
in while 1 to 10 do
```

```
let y: α := ~
in if i = 5 then
  x := &y
```

```
! let z: α := *x
```

↑
output-value has
also not writ
but is not loc.
typesafe