COMA: an intermediate verification language with explicit abstraction barriers

Andrei Paskevich and Paul Patault

with thanks to Jean-Christophe Filliâtre

LMF, Université Paris-Saclay • Toccata, Inria Saclay

Cambium seminar • 9 oct 2023

$$
\begin{aligned}
\mathrm{WP}(\texttt{skip}, \Phi) &\triangleq \Phi \\
\mathrm{WP}(e \mathbin{;} d, \Phi) &\triangleq \mathrm{WP}(e, \mathrm{WP}(d, \Phi)) \\
\mathrm{WP}(x \leftarrow v, \Phi) &\triangleq \Phi[x \mapsto v] \\
\mathrm{WP}(\texttt{if } \varphi \texttt{ then } e \texttt{ else } d, \Phi) &\triangleq \texttt{if } \varphi \texttt{ then } \mathrm{WP}(e, \Phi) \texttt{ else } \mathrm{WP}(d, \Phi) \\
\mathrm{WP}(\texttt{while } \varphi \texttt{ do } e \texttt{ done}, \Phi) &\triangleq \exists \psi.\, \psi \wedge \forall \bar{x}.\, \psi \longrightarrow \texttt{if } \varphi \texttt{ then } \mathrm{WP}(e, \psi) \\
&\qquad\qquad\qquad\qquad\quad \texttt{else } \Phi
\end{aligned}
$$

$$\mathrm{WP}(\texttt{skip}, \Phi, \Psi) \triangleq \Phi$$

$$\mathrm{WP}(e\ ;d, \Phi, \Psi) \triangleq \mathrm{WP}(e, \mathrm{WP}(d, \Phi, \Psi), \Psi)$$

$$\begin{aligned}
\mathrm{WP}(\texttt{skip}, \Phi, \Psi) &\triangleq \Phi \\
\mathrm{WP}(e \,;\, d, \Phi, \Psi) &\triangleq \mathrm{WP}(e, \mathrm{WP}(d, \Phi, \Psi), \Psi) \\
\\
\mathrm{WP}(\texttt{raise}, \Phi, \Psi) &\triangleq \Psi \\
\mathrm{WP}(\texttt{try}\ e\ \texttt{with}\ d, \Phi, \Psi) &\triangleq \mathrm{WP}(e, \Phi, \mathrm{WP}(d, \Phi, \Psi))
\end{aligned}$$

$$
\begin{array}{rcll}
x, y, z & & & \text{variable} \\
s, t & ::= & x \mid 0 \ldots \mid s + t \ldots & \text{term} \\
\varphi, \psi & ::= & s = t \ldots \mid \varphi \wedge \psi \ldots & \text{formula}
\end{array}
$$

data

$$x, y, z \qquad\qquad \text{variable}$$

$$s, t \quad ::= \quad x \mid 0 \ldots \mid s + t \ldots \qquad\qquad \text{term}$$

$$\varphi, \psi \quad ::= \quad s = t \ldots \mid \varphi \wedge \psi \ldots \qquad\qquad \text{formula}$$

data

---

code

$$h, g, f \qquad\qquad \text{handler}$$

$$\pi, \varrho \quad ::= \quad h \;\; \bar{x} \;\; \varphi \;\; \bar{\pi} \qquad\qquad \text{prototype}$$

$$
\begin{array}{rcll}
x, y, z & & & \text{variable} \\
s, t & ::= & x \mid 0 \ldots \mid s + t \ldots & \text{term} \\
\varphi, \psi & ::= & s = t \ldots \mid \varphi \wedge \psi \ldots & \text{formula}
\end{array}
$$

data

code

$$
\begin{array}{rcll}
h, g, f & & & \text{handler} \\
\pi, \varrho & ::= & h \;\; \bar{x} \;\; \varphi \;\; \bar{\pi} & \text{prototype} \\
e, d & ::= & h \;\; \bar{s} \;\; \bar{g} & \text{expression} \\
 & \mid & e \;\; / \;\; h \;\; \bar{x} \;\; \varphi \;\; \bar{\pi} \;\; = \;\; d & \\
 & \mid & e \;\; / \;\; h \;\; \bar{x} \;\; \bar{\pi} \;\; \rightarrow \;\; d &
\end{array}
$$

```
factorial (n: int) { n ⩾ 0 }
  (return (m: int) { m = n! })
= fact 1 n
  / fact (r: int) (k: int)
      { 0 ⩽ k ⩽ n ∧ r · k! = n! }
    = if (k > 0) tt ff
      / tt → fact (r * k) (k - 1)
      / ff → return r
```

user-written code

predefined handlers

```
if (b: bool) (then { b }) (else { ¬b })
halt { ⊤ }
fail { ⊥ }
```

$$\mathrm{WP}(e \;/\; h \; \bar{x} \; \varphi \; \bar{\pi} = d) \;\triangleq\; \mathrm{WP}(e) \wedge \forall \bar{x}.\, \varphi \longrightarrow \mathrm{WP}(d)$$

$\mathrm{WP}(\textsc{Coma}_0)$

$$\mathrm{WP}(e \mathbin{/} h\ \bar{x}\ \bar{\pi} \rightarrow d) \triangleq \mathrm{WP}(e)$$

$$\mathrm{WP}(e \mathbin{/} h\ \bar{x}\ \varphi\ \bar{\pi} = d) \triangleq \mathrm{WP}(e) \land \forall \bar{x}.\, \varphi \longrightarrow \mathrm{WP}(d)$$

$$\mathrm{WP}(h_{\bar{x}\bar{\pi}\to d}\ \bar{s}\ \bar{g}) \triangleq \mathrm{WP}(d)[\bar{x} \mapsto \bar{s}] \wedge \bigwedge_{i=1}^{n} (\pi_i[\bar{x} \mapsto \bar{s}] \blacktriangleright g_i)$$

$$\mathrm{WP}(e\ /\ h\ \bar{x}\ \bar{\pi} \to d) \triangleq \mathrm{WP}(e)$$

$$\mathrm{WP}(e\ /\ h\ \bar{x}\ \varphi\ \bar{\pi} = d) \triangleq \mathrm{WP}(e) \wedge \forall \bar{x}. \varphi \longrightarrow \mathrm{WP}(d)$$

$$\mathrm{WP}(h_{\bar{x}\bar{\pi}\to d}\ \bar{s}\ \bar{g}) \triangleq \mathrm{WP}(d)[\bar{x} \mapsto \bar{s}] \wedge \bigwedge_{i=1}^{n}(\pi_i[\bar{x} \mapsto \bar{s}] \blacktriangleright g_i)$$

$$\mathrm{WP}(e\ /\ h\ \bar{x}\ \bar{\pi} \to d) \triangleq \mathrm{WP}(e)$$

$$\mathrm{WP}(e\ /\ h\ \bar{x}\ \varphi\ \bar{\pi} = d) \triangleq \mathrm{WP}(e) \wedge \forall \bar{x}.\varphi \longrightarrow \mathrm{WP}(d)$$

$$h\ \bar{x}\ \varphi\ \bar{\pi} \blacktriangleright g \triangleq \forall \bar{x}.\varphi \longrightarrow \mathrm{WP}(g\ \bar{x}\ \bar{\pi})$$

WP(ComA$_0$)

$$\mathrm{WP}(h_{\bar{x}\varphi\bar{\pi}}\ \bar{s}\ \bar{g}) \triangleq \varphi[\bar{x} \mapsto \bar{s}] \wedge \bigwedge_{i=1}^{n} (\pi_i[\bar{x} \mapsto \bar{s}] \blacktriangleright g_i)$$

$$\mathrm{WP}(h_{\bar{x}\bar{\pi}\to d}\ \bar{s}\ \bar{g}) \triangleq \mathrm{WP}(d)[\bar{x} \mapsto \bar{s}] \wedge \bigwedge_{i=1}^{n} (\pi_i[\bar{x} \mapsto \bar{s}] \blacktriangleright g_i)$$

$$\mathrm{WP}(e\ /\ h\ \bar{x}\ \bar{\pi} \to d) \triangleq \mathrm{WP}(e)$$

$$\mathrm{WP}(e\ /\ h\ \bar{x}\ \varphi\ \bar{\pi} = d) \triangleq \mathrm{WP}(e) \wedge \forall \bar{x}.\varphi \longrightarrow \mathrm{WP}(d)$$

$$h\ \bar{x}\ \varphi\ \bar{\pi} \blacktriangleright g \triangleq \forall \bar{x}.\varphi \longrightarrow \mathrm{WP}(g\ \bar{x}\ \bar{\pi})$$

```
factorial (n: int) { n ⩾ 0 }
  (return (m: int) { m = n! })
= fact 1 n
  / fact (r: int) (k: int)
      { 0 ⩽ k ⩽ n ∧ r · k! = n! }
    = if (k > 0) tt ff
      / tt → fact (r * k) (k - 1)
      / ff → return r
```

$$
\begin{aligned}
&\forall n: \mathsf{int}.\, n \geqslant 0 \;\rightarrow\; 0 \leqslant n \leqslant n \;\wedge\; 1 \cdot n! = n! \;\wedge \\
&\qquad \forall r: \mathsf{int}.\, \forall k: \mathsf{int}.\, 0 \leqslant k \leqslant n \;\wedge\; r \cdot k! = n! \;\rightarrow \\
&\qquad\qquad (k > 0 \;\rightarrow\; 0 \leqslant k - 1 \leqslant n \;\wedge\; r \cdot k \cdot (k-1)! = n!) \;\wedge \\
&\qquad\qquad (k \leqslant 0 \;\rightarrow\; r = n!)
\end{aligned}
$$

```
type tree = Node tree elt tree
          | Leaf

let removeRoot (t: tree) : tree




= match t with
  | Node l _ r → mergeTree l r
  | Leaf → fail
```

```
type tree = Node tree elt tree
          | Leaf

let removeRoot (t: tree) : tree
  requires { t ≠ Leaf }
  ensures  { match t with
    | Node l _ r → ∀e:elt. e ∈ result ↔ e ∈ l ∨ e ∈ r
    | Leaf → false }
= match t with
  | Node l _ r → mergeTree l r
  | Leaf → fail
```

```
type tree = Node tree elt tree
          | Leaf

removeRoot (Node l _ r)
  ensures { ∀e:elt. e ∈ result ↔ e ∈ l ∨ e ∈ r }
= mergeTree l r

removeRoot Leaf = fail
```

h, g, f			handler
π, ϱ	$::=$	$h \;\; \bar{x} \;\; \bar{\pi}$	prototype
k, o	$::=$	h	callable
	$\mid$	$\lambda \;\; \bar{x} \;\; \bar{\pi} \;\; . \;\; e$	
e, d	$::=$	$k \;\; \bar{s} \;\; \bar{o}$	expression
	$\mid$	$e \;\; / \;\; h \;\; \bar{x} \;\; \bar{\pi} \;\; = \;\; d$	
	$\mid$	$\varphi \;\; e$	
	$\mid$	$\uparrow \;\; e$	

```
factorial (n: int) (return (m: int))
= { n ⩾ 0 }
  ↑ fact 1 n
    / fact (r: int) (k: int)
      = { 0 ⩽ k ⩽ n ∧ r · k! = n! }
        ↑ if (k > 0) (λ. fact (r * k) (k - 1))
                     (λ. break r)
  / break (m: int) = { m = n! } ↑ return m
```

```
removeRoot (t: tree) (return (s: tree))
= unTree t (λl: tree. λ_: elt. λr: tree.
              ↑ mergeTree l r out
              / out (s: tree) =
                  { ∀e:elt. e ∈ s ↔ e ∈ l ∨ e ∈ r }
                  ↑ return s)
           fail
```

or none at all

```
removeRoot (t: tree) (return (s: tree))
= unTree t (λl: tree. λ_: elt. λr: tree.
              mergeTree l r return)
           fail
```

$\mathbb{C}^{\mathfrak{p}}_{\mathfrak{d}}(e)$, $\mathbb{C}^{\mathfrak{p}}_{\mathfrak{d}}(k)$ — verification condition of an expression or a callable

- $\mathfrak{p}$ — should we produce proof obligations at the current position?
- $\mathfrak{d}$ — should we produce proof obligations after the barrier $\uparrow$?

$\mathbb{C}^{\top}_{\bot}$ — caller VC, the specification (contract) of a handler

$\mathbb{C}^{\bot}_{\top}$ — callee VC, implementation respects the contract

$\mathbb{C}^{\top}_{\top}$ — full VC, prove everything, ignore the barriers

$\mathbb{C}^{\bot}_{\bot}$ — null VC, prove nothing, ignore the barriers

$$\mathsf{C}^{\top}_{\mathfrak{d}}(h) \triangleq h \qquad \mathsf{C}^{\bot}_{\mathfrak{d}}(h) \triangleq \natural h$$

$$\natural\Phi \longrightarrow \Phi[\overline{f \mapsto \natural f}]_{f \in \mathrm{FS}(\Phi)}$$

$$\natural\mathtt{fail} \longrightarrow \top$$

$$\mathsf{C}^{\top}_{\mathfrak{d}}(h) \triangleq h \qquad\qquad \mathsf{C}^{\bot}_{\mathfrak{d}}(h) \triangleq \natural h$$

$$\mathsf{C}^{\mathfrak{p}}_{\mathfrak{d}}(\varphi\ e) \triangleq \texttt{if}\ \varphi\ \texttt{then}\ \mathsf{C}^{\mathfrak{p}}_{\mathfrak{d}}(e)\ \texttt{else}\ \mathsf{C}^{\mathfrak{p}}_{\mathfrak{d}}(\texttt{fail})$$

$$\natural\Phi \longrightarrow \Phi[\overline{f \mapsto \natural f}]_{f \in \mathrm{FS}(\Phi)}$$

$$\natural\texttt{fail} \longrightarrow \top$$

$$\mathbb{C}^{\top}_{\mathfrak{d}}(h) \triangleq h \qquad \mathbb{C}^{\bot}_{\mathfrak{d}}(h) \triangleq \natural h$$

$$\begin{aligned}
\mathbb{C}^{\mathfrak{p}}_{\mathfrak{d}}(e \,/\, h\ \bar{x}\ \bar{\pi} = d) &\triangleq \texttt{let}\ h = \lambda \bar{x}\bar{\pi}.\quad \mathbb{C}^{\top}_{\bot}(d) \\
&\qquad \texttt{in}\ \mathbb{C}^{\mathfrak{p}}_{\mathfrak{d}}(e) \wedge \forall \bar{x}\bar{\pi}.\, \mathbb{C}^{\bot}_{\mathfrak{p}}(d) \\
\mathbb{C}^{\mathfrak{p}}_{\mathfrak{d}}(\varphi\ e) &\triangleq \texttt{if}\ \varphi\ \texttt{then}\ \mathbb{C}^{\mathfrak{p}}_{\mathfrak{d}}(e)\ \texttt{else}\ \mathbb{C}^{\mathfrak{p}}_{\mathfrak{d}}(\texttt{fail})
\end{aligned}$$

$$\natural\Phi \longrightarrow \Phi[\overline{f \mapsto \natural f}]_{f \in \mathrm{FS}(\Phi)}$$

$$\natural\texttt{fail} \longrightarrow \top$$

$$\mathcal{C}^{\top}_{\mathfrak{d}}(h) \triangleq h \qquad \mathcal{C}^{\perp}_{\mathfrak{d}}(h) \triangleq \natural h$$

$$\mathcal{C}^{\mathfrak{p}}_{\mathfrak{d}}(e \,/\, h\ \bar{x}\ \bar{\pi} = d) \triangleq \texttt{let}\ h = \lambda \bar{x}\bar{\pi}. \forall h. \mathcal{C}^{\top}_{\perp}(d)\ \texttt{in}\ \mathcal{C}^{\mathfrak{p}}_{\mathfrak{d}}(e) \wedge \forall \bar{x}\bar{\pi}. \mathcal{C}^{\perp}_{\mathfrak{p}}(d)$$

$$\mathcal{C}^{\mathfrak{p}}_{\mathfrak{d}}(\varphi\ e) \triangleq \texttt{if}\ \varphi\ \texttt{then}\ \mathcal{C}^{\mathfrak{p}}_{\mathfrak{d}}(e)\ \texttt{else}\ \mathcal{C}^{\mathfrak{p}}_{\mathfrak{d}}(\texttt{fail})$$

$$\natural\Phi \longrightarrow \Phi[\overline{f \mapsto \natural f}]_{f \in \mathrm{FS}(\Phi)}$$

$$\natural\texttt{fail} \longrightarrow \top$$

$$\forall h_{\bar{x}\bar{\pi}}. \Phi \triangleq \texttt{let}\ h = \lambda \bar{x}\bar{\pi}. \texttt{fail} \wedge \bigwedge_{g_{\bar{z}\bar{\varrho}} \in \bar{\pi}} \forall \bar{z}\bar{\varrho}. g\ \bar{z}\ \bar{\varrho}\ \texttt{in}\ \Phi$$

$$\mathbb{C}^{\top}_{\mathfrak{d}}(h) \triangleq h \qquad\qquad \mathbb{C}^{\bot}_{\mathfrak{d}}(h) \triangleq \natural h$$

$$\begin{aligned}
\mathbb{C}^{\mathfrak{p}}_{\mathfrak{d}}(e \,/\, h\ \bar{x}\ \bar{\pi} = d) &\triangleq \texttt{let}\ h = \lambda \bar{x}\bar{\pi}.\forall h.\, \mathbb{C}^{\top}_{\bot}(d) \\
&\qquad \texttt{in}\ \mathbb{C}^{\mathfrak{p}}_{\mathfrak{d}}(e) \wedge \forall \bar{x}\bar{\pi}.\, \mathbb{C}^{\bot}_{\mathfrak{p}}(d) \\
\mathbb{C}^{\mathfrak{p}}_{\mathfrak{d}}(\varphi\ e) &\triangleq \texttt{if}\ \varphi\ \texttt{then}\ \mathbb{C}^{\mathfrak{p}}_{\mathfrak{d}}(e)\ \texttt{else}\ \mathbb{C}^{\mathfrak{p}}_{\mathfrak{d}}(\texttt{fail}) \\
\mathbb{C}^{\mathfrak{p}}_{\mathfrak{d}}(\uparrow e) &\triangleq \mathbb{C}^{\mathfrak{d}}_{\mathfrak{d}}(e)
\end{aligned}$$

$$\natural\Phi \longrightarrow \Phi[\overline{f \mapsto \natural f}]_{f \in \mathrm{FS}(\Phi)}$$

$$\natural\texttt{fail} \longrightarrow \top$$

$$\forall h_{\bar{x}\bar{\pi}}.\Phi \triangleq \texttt{let}\ h = \lambda \bar{x}\bar{\pi}.\ \texttt{fail} \wedge \bigwedge_{g_{\bar{z}\bar{\varrho}} \in \bar{\pi}} \forall \bar{z}\bar{\varrho}.\, g\ \bar{z}\ \bar{\varrho}\ \texttt{in}\ \Phi$$

$$\mathcal{C}^{\top}_{\mathfrak{d}}(h) \triangleq h \qquad \mathcal{C}^{\perp}_{\mathfrak{d}}(h) \triangleq \natural h$$

$$\mathcal{C}^{\mathfrak{p}}_{\mathfrak{d}}(k\ \bar{s}\ \bar{o}) \triangleq \mathcal{C}^{\mathfrak{p}}_{\mathfrak{d}}(k)\ \bar{s}\ \mathcal{C}^{\mathfrak{p}}_{\mathfrak{d}}(o_1)\ \cdots\ \mathcal{C}^{\mathfrak{p}}_{\mathfrak{d}}(o_n)$$

$$\mathcal{C}^{\mathfrak{p}}_{\mathfrak{d}}(e\ /\ h\ \bar{x}\ \bar{\pi} = d) \triangleq \texttt{let}\ h = \lambda \bar{x}\bar{\pi}. \forall h. \mathcal{C}^{\top}_{\perp}(d)$$

$$\texttt{in}\ \mathcal{C}^{\mathfrak{p}}_{\mathfrak{d}}(e) \wedge \forall \bar{x}\bar{\pi}. \mathcal{C}^{\perp}_{\mathfrak{p}}(d)$$

$$\mathcal{C}^{\mathfrak{p}}_{\mathfrak{d}}(\varphi\ e) \triangleq \texttt{if}\ \varphi\ \texttt{then}\ \mathcal{C}^{\mathfrak{p}}_{\mathfrak{d}}(e)\ \texttt{else}\ \mathcal{C}^{\mathfrak{p}}_{\mathfrak{d}}(\texttt{fail})$$

$$\mathcal{C}^{\mathfrak{p}}_{\mathfrak{d}}(\uparrow e) \triangleq \mathcal{C}^{\mathfrak{d}}_{\mathfrak{d}}(e)$$

$$\natural\Phi \longrightarrow \Phi[\overline{f \mapsto \natural f}]_{f \in \mathrm{FS}(\Phi)}$$

$$\natural\texttt{fail} \longrightarrow \top$$

$$\forall h_{\bar{x}\bar{\pi}}.\Phi \triangleq \texttt{let}\ h = \lambda \bar{x}\bar{\pi}.\ \texttt{fail} \wedge \bigwedge_{g_{\bar{z}\bar{\varrho}} \in \bar{\pi}} \forall \bar{z}\bar{\varrho}. g\ \bar{z}\ \bar{\varrho}\ \texttt{in}\ \Phi$$

VC(COMA)

$$
\begin{aligned}
\mathcal{C}^{\top}_{\mathfrak{d}}(h) &\triangleq h \qquad\qquad \mathcal{C}^{\bot}_{\mathfrak{d}}(h) \triangleq \natural h \\
\mathcal{C}^{\mathfrak{p}}_{\mathfrak{d}}(\lambda \bar{x}\bar{\pi}.\, e) &\triangleq (\lambda \bar{x}\bar{\pi}.\, \mathcal{C}^{\mathfrak{p}}_{\mathfrak{d}}(e)) \wedge \natural(\lambda \bar{x}\bar{\pi}.\, \mathcal{C}^{\neg\mathfrak{p}}_{\neg\mathfrak{d}}(e)) \\
\mathcal{C}^{\mathfrak{p}}_{\mathfrak{d}}(k\ \bar{s}\ \bar{o}) &\triangleq \mathcal{C}^{\mathfrak{p}}_{\mathfrak{d}}(k)\ \bar{s}\ \mathcal{C}^{\mathfrak{p}}_{\mathfrak{d}}(o_1)\ \cdots\ \mathcal{C}^{\mathfrak{p}}_{\mathfrak{d}}(o_n) \\
\mathcal{C}^{\mathfrak{p}}_{\mathfrak{d}}(e\ /\ h\ \bar{x}\ \bar{\pi} = d) &\triangleq \mathtt{let}\ h = \lambda \bar{x}\bar{\pi}.\forall h.\, \mathcal{C}^{\top}_{\bot}(d) \\
&\qquad \mathtt{in}\ \mathcal{C}^{\mathfrak{p}}_{\mathfrak{d}}(e) \wedge \forall \bar{x}\bar{\pi}.\, \mathcal{C}^{\bot}_{\mathfrak{p}}(d) \\
\mathcal{C}^{\mathfrak{p}}_{\mathfrak{d}}(\varphi\ e) &\triangleq \mathtt{if}\ \varphi\ \mathtt{then}\ \mathcal{C}^{\mathfrak{p}}_{\mathfrak{d}}(e)\ \mathtt{else}\ \mathcal{C}^{\mathfrak{p}}_{\mathfrak{d}}(\mathtt{fail}) \\
\mathcal{C}^{\mathfrak{p}}_{\mathfrak{d}}(\uparrow e) &\triangleq \mathcal{C}^{\mathfrak{d}}_{\mathfrak{d}}(e)
\end{aligned}
$$

$$\natural\Phi \longrightarrow \Phi[\overline{f \mapsto \natural f}]_{f\in \mathrm{FS}(\Phi)} \qquad (\Phi \wedge \Psi)\ s \longrightarrow \Phi\ s \wedge \Psi\ s$$

$$\natural\mathtt{fail} \longrightarrow \top \qquad (\Phi \wedge \Psi)\ \Upsilon \longrightarrow \Phi\ \Upsilon \wedge \Psi\ \Upsilon$$

$$\forall h_{\bar{x}\bar{\pi}}.\Phi \triangleq \mathtt{let}\ h = \lambda \bar{x}\bar{\pi}.\ \mathtt{fail} \wedge \bigwedge_{g_{\bar{z}\bar{\varrho}} \in \bar{\pi}} \forall \bar{z}\bar{\varrho}.\, g\ \bar{z}\ \bar{\varrho}\ \mathtt{in}\ \Phi$$

VC(COMA)

$$\mathcal{C}^{\top}_{\mathfrak{d}}(h) \triangleq h \qquad \mathcal{C}^{\perp}_{\mathfrak{d}}(h) \triangleq \natural h$$

$$\mathcal{C}^{\mathfrak{p}}_{\mathfrak{d}}(\lambda \bar{x}\bar{\pi}.\, e) \triangleq (\lambda \bar{x}\bar{\pi}.\, \mathcal{C}^{\mathfrak{p}}_{\mathfrak{d}}(e)) \wedge \natural(\lambda \bar{x}\bar{\pi}.\, \mathcal{C}^{\neg\mathfrak{p}}_{\neg\mathfrak{d}}(e))$$

$$\mathcal{C}^{\mathfrak{p}}_{\mathfrak{d}}(k\ \bar{s}\ \bar{o}) \triangleq \mathcal{C}^{\mathfrak{p}}_{\mathfrak{d}}(k)\ \bar{s}\ \mathcal{C}^{\mathfrak{p}}_{\mathfrak{d}}(o_1) \cdots \mathcal{C}^{\mathfrak{p}}_{\mathfrak{d}}(o_n)$$

$$\mathcal{C}^{\mathfrak{p}}_{\mathfrak{d}}(e\ /\ h\ \bar{x}\ \bar{\pi} = d) \triangleq \texttt{let}\ h = \lambda \bar{x}\bar{\pi}. \forall h.\, \mathcal{C}^{\top}_{\perp}(d)\ \texttt{in}\ \mathcal{C}^{\mathfrak{p}}_{\mathfrak{d}}(e) \wedge \forall \bar{x}\bar{\pi}.\, \mathcal{C}^{\perp}_{\mathfrak{p}}(d)$$

$$\mathcal{C}^{\mathfrak{p}}_{\mathfrak{d}}(\varphi\ e) \triangleq \texttt{if}\ \varphi\ \texttt{then}\ \mathcal{C}^{\mathfrak{p}}_{\mathfrak{d}}(e)\ \texttt{else}\ \mathcal{C}^{\mathfrak{p}}_{\mathfrak{d}}(\texttt{fail})$$

$$\mathcal{C}^{\mathfrak{p}}_{\mathfrak{d}}(\uparrow e) \triangleq \mathcal{C}^{\mathfrak{d}}_{\mathfrak{d}}(e) \qquad \mathcal{C}^{\mathfrak{p}}_{\mathfrak{d}}(\downarrow e) \triangleq \mathcal{C}^{\mathfrak{p}}_{\mathfrak{p}}(e)$$

$$\natural\Phi \longrightarrow \Phi[\overline{f \mapsto \natural f}]_{f \in \mathrm{FS}(\Phi)} \qquad (\Phi \wedge \Psi)\ s \longrightarrow \Phi\ s \wedge \Psi\ s$$

$$\natural\texttt{fail} \longrightarrow \top \qquad (\Phi \wedge \Psi)\ \Upsilon \longrightarrow \Phi\ \Upsilon \wedge \Psi\ \Upsilon$$

$$\forall h_{\bar{x}\bar{\pi}}.\, \Phi \triangleq \texttt{let}\ h = \lambda \bar{x}\bar{\pi}.\ \texttt{fail} \wedge \bigwedge_{g_{\bar{z}\bar{\varrho}} \in \bar{\pi}} \forall \bar{z}\bar{\varrho}.\, g\ \bar{z}\ \bar{\varrho}\ \texttt{in}\ \Phi$$

mutable state

p, q, r			reference
π, ϱ	$::=$	$h\ [\bar{q}]\ \&\bar{p}\ \ \bar{x}\ \ \bar{\pi}$	prototype
k, o	$::=$	h	callable
	$\mid$	$\lambda\ \ \&\bar{p}\ \ \bar{x}\ \ \bar{\pi}\ .\ e$	
e, d	$::=$	$k\ \ \&\bar{r}\ \ \bar{s}\ \ \bar{o}$	expression
	$\mid$	$e\ /\ h\ [\bar{q}]\ \&\bar{p}\ \ \bar{x}\ \ \bar{\pi}\ =\ d$	
	$\mid$	$\varphi\ e$	
	$\mid$	$\uparrow\ e$	

$h\ [\bar{q}]$ — pre-write annotation, $\bar{q}$ may be modified before h is called

```
factorial (n: int) (return (m: int))
= { n ⩾ 0 }
  allocate 1 (λ&r: int.
    ↑ allocate n (λ&k: int.
        fact
        / fact [r k]
          = { 0 ⩽ k ⩽ n ∧ r · k! = n! }
            ↑ if (k > 0) (λ. assign &r (r ∗ k)
                           (λ. assign &k (k - 1)
                             (λ. fact)))
                         (λ. break))
    / break [r] = { r = n! } ↑ return r)
```

```
allocate (v: int) (return (&r: int) { r = v })
assign (&r: int) (v: int) (return [r] { r = v })
```

```
factorial (n: int) (return (m: int))
= { n ⩾ 0 }
  allocate int 1 (λ&r: int.
    ↑ allocate int n (λ&k: int.
        fact
        / fact [r k]
          = { 0 ⩽ k ⩽ n ∧ r · k! = n! }
            ↑ if (k > 0) (λ. assign int &r (r ∗ k)
                              (λ. assign int &k (k - 1)
                                (λ. fact)))
                            (λ. break))
    / break [r] = { r = n! } ↑ return r)
```

```
allocate α (v: α) (return (&r: α) { r = v })
assign α (&r: α) (v: α) (return [r] { r = v })
```

No-alias type system:

$$\frac{\Gamma, \Delta' \vdash e \ \mathsf{wt} \qquad \Delta' \textit{ is } \Delta \textit{ with all handler prototypes removed}}{\Gamma, \&r, \Delta \vdash (e \ \&r) \ \mathsf{wt}}$$

– can be further refined by tracking actual reference dependencies

Effect computation — to verify and infer the pre-write annotations

No-alias type system:

$$\frac{\Gamma, \Delta' \vdash e \ \mathsf{wt} \qquad \Delta' \textit{ is } \Delta \textit{ with all handler prototypes removed}}{\Gamma, \&r, \Delta \vdash (e\ \&r)\ \mathsf{wt}}$$

– can be further refined by tracking actual reference dependencies

Effect computation — to verify and infer the pre-write annotations

Transformation into an equivalent pure program:

```
assign &r (r * k)                 assign r (r * k)
  (λ. assign &k (k - 1)     ⇒       (λr′. assign k (k - 1)
    (λ. fact))                         (λk′. fact r′ k′))
```

– pre-writes are converted into term parameters

– fine-grained state monad: send only the relevant part of the state

Krivine-style evaluator over COMA expressions — no intermediate HOL

A few ways to speed up the computation, e.g. $\complement^{\perp}_{\perp}(e) \equiv \natural\complement^{\mathfrak{p}}_{\mathfrak{d}}(e) \equiv \top$

On-the-fly factorization of selected handlers:

$$(\forall x.\,\varphi \to h\ s) \land (\forall y.\,\psi \to h\ t) \quad\Longrightarrow$$
$$\forall z.\,((\exists x.\,\varphi \land z = s) \lor (\exists y.\,\psi \land z = t)) \to h\ z$$

- no factorized handlers $\approx$ traditional WP
- factorize all eligible handlers $\approx$ compact VC à la Flanagan & Saxe

Further into control structures: iterators, coroutines, unstructured code?

Further into mutable state: ownership, borrowing, prophecy variables?

Scalable implementation, good heuristics for subgoal factorization

Nice surface syntax, extensive case studies, integration into WHY3