Vocabulary of the trustful JVM$_{\mathcal{O}}$

Instructions:

$$
\begin{aligned}
\mathit{Instr} = {} & \ldots \\
& \mid \mathit{New}(\mathit{Class}) \\
& \mid \mathit{GetField}(\mathit{Type}, \mathit{Class}/\mathit{Field}\,) \\
& \mid \mathit{PutField}(\mathit{Type}, \mathit{Class}/\mathit{Field}\,) \\
& \mid \mathit{InstanceOf}(\mathit{Type}) \\
& \mid \mathit{Checkcast}(\mathit{Type}) \\
& \mid \mathit{InvokeSpecial}(\mathit{Type}, \mathit{Class}/\mathit{MSig}) \\
& \mid \mathit{InvokeVirtual}(\mathit{Type}, \mathit{Class}/\mathit{MSig})
\end{aligned}
$$

Universes:

$$
\begin{aligned}
\mathit{MoveType} = {} & \ldots \\
& \mid \mathtt{addr}
\end{aligned}
$$

Trustful execution of JVM$_\mathcal{O}$ instructions

```
$execVM_O(instr) =$
  $execVM_C(instr)$
  case $instr$ of
    $New(c) \rightarrow$
      if $initialized(c)$ then create $r$
        $heap(r) := Object(c, \{(f, defaultVal(f)) \mid f \in instFields(c)\})$
        $opd := opd \cdot [r]$
        $pc \quad := pc + 1$
      else $switch := InitClass(c)$
```

Trustful execution of JVM$_{\mathcal{O}}$ instructions (continued)

```
$execVM_O(instr) =$
  case $instr$ of
    $GetField(\_, c/f) \rightarrow$ let $(opd', [r]) = split(opd, 1)$
                        if $r \neq null$ then
                          $opd := opd' \cdot getField(r, c/f)$
                          $pc \ := pc + 1$
    $PutField(\_, c/f) \rightarrow$ let $(opd', [r] \cdot ws) = split(opd, 1 + size(c/f))$
                        if $r \neq null$ then
                          $setField(r, c/f, ws)$
                          $pc \ := pc + 1$
                          $opd := opd'$
```

Trustful execution of JVM$_\mathcal{O}$ instructions (continued)

```
$execVM_O(instr) =$
  **case** $instr$ **of**
    $InvokeSpecial(\_, c/m) \rightarrow$
      **let** $(opd', [r] \cdot ws) = split(opd, 1 + argSize(c/m))$
      **if** $r \neq null$ **then**
        $opd \quad := opd'$
        $switch := Call(c/m, [r] \cdot ws)$
    $InvokeVirtual(\_, c/m) \rightarrow$
      **let** $(opd', [r] \cdot ws) = split(opd, 1 + argSize(c/m))$
      **if** $r \neq null$ **then**
        $opd \quad := opd'$
        $switch := Call(lookup(classOf(r), c/m), [r] \cdot ws)$
```

Trustful execution of JVM$_\mathcal{O}$ instructions (continued)

```
$execVM_O(instr) =$
  **case** $instr$ **of**
    $InstanceOf(c) \rightarrow$ **let** $(opd', [r]) = split(opd, 1)$
                          $opd := opd' \cdot (r \neq null \wedge classOf(r) \sqsubseteq c)$
                          $pc := pc + 1$
    $Checkcast(c) \rightarrow$ **let** $r = top(opd)$
                          **if** $r = null \vee classOf(r) \sqsubseteq c$ **then**
                            $pc := pc + 1$
```

$$trustfulVM_O = trustfulScheme_C(execVM_O, switchVM_C)$$

Compilation of Java$_\mathcal{O}$ expressions

$$
\begin{aligned}
\mathcal{E}(\texttt{this}) &= \mathit{Load}(\texttt{addr}, 0)\\
\mathcal{E}(\texttt{new}\ c) &= \mathit{New}(c) \cdot \mathit{Dupx}(0, 1)\\
\mathcal{E}(\mathit{exp}.c/f) &= \mathcal{E}(\mathit{exp}) \cdot \mathit{GetField}(\mathcal{T}(c/f), c/f)\\
\mathcal{E}(\mathit{exp}_1.c/f = \mathit{exp}_2) &= \mathcal{E}(\mathit{exp}_1) \cdot \mathcal{E}(\mathit{exp}_2) \cdot \mathit{Dupx}(1, \mathit{size}(\mathcal{T}(c/f)))\cdot\\
&\ \mathit{PutField}(\mathcal{T}(c/f), c/f)\\
\mathcal{E}(\mathit{exp}.c/m(\mathit{exps})) &= \mathcal{E}(\mathit{exp}) \cdot \mathcal{E}(\mathit{exps})\cdot\\
&\ \textbf{case}\ \mathit{callKind}(\mathit{exp}.c/m)\ \textbf{of}\\
&\ \quad \mathit{Virtual} \rightarrow \mathit{InvokeVirtual}(\mathcal{T}(c/m), c/m)\\
&\ \quad \mathit{Super} \rightarrow \mathit{InvokeSpecial}(\mathcal{T}(c/m), c/m)\\
&\ \quad \mathit{Special} \rightarrow \mathit{InvokeSpecial}(\mathcal{T}(c/m), c/m)\\
\\
\mathcal{E}(\mathit{exp}\ \texttt{instanceof}\ c) &= \mathcal{E}(\mathit{exp}) \cdot \mathit{InstanceOf}(c)\\
\mathcal{E}((c)\mathit{exp}) &= \mathcal{E}(\mathit{exp}) \cdot \mathit{Checkcast}(c)
\end{aligned}
$$