1 Grammar

In our calculus we denote set of methods in a program by M and the set of resources by R. Elements of those sets are denoted m and r respectively. An effect ε is a member of the set of pairs $M \times R$. Intuitively we may read the effect (m, r) as meaning 'the effect on resource r when method m is called'.

$$\begin{array}{rcll}
e & ::= & x & \textit{expressions} \\
 & | & \texttt{new}\ x \Rightarrow \overline{\sigma = e} & \\
 & | & e.m(e) & \\
 & | & r & \\
 & & & \\
\tau & ::= & \{\bar{\sigma}\} \mid \{\bar{r}\} & \textit{types} \\
 & & & \\
d & ::= & \texttt{def}\ m(x : \tau) : \tau & \textit{declarations} \\
 & & & \\
\sigma & ::= & d\ \texttt{with}\ \varepsilon & \textit{annotated decls.}
\end{array}$$

2 Effect Rules (Green)

$$\frac{}{\Gamma, x : \tau \vdash x : \tau\ \texttt{with}\ \varnothing}\ (\varepsilon\text{-Var}) \qquad \frac{}{\Gamma \vdash r : \{r\}\ \texttt{with}\ \varnothing}\ (\varepsilon\text{-Resource})$$

$$\frac{\Gamma, x : \tau \vdash e : \tau'\ \texttt{with}\ \varepsilon \quad \sigma = \texttt{def}\ m(x : \tau) : \tau'\ \texttt{with}\ \varepsilon}{\Gamma \vdash \sigma = e\ \texttt{OK}}\ (\varepsilon\text{-ValidImpl})$$

$$\frac{\Gamma, x : \{\bar{\sigma}\} \vdash \overline{\sigma = e}\ \texttt{OK}}{\Gamma \vdash \texttt{new}\ x \Rightarrow \overline{\sigma = e} : \{\bar{\sigma}\}\ \texttt{with}\ \varnothing}\ (\varepsilon\text{-NewObj})$$

$$\frac{\Gamma \vdash e_1 : \{\bar{r}\}\ \texttt{with}\ \varepsilon_1 \quad \Gamma \vdash e_2 : \tau_2\ \texttt{with}\ \varepsilon_2}{\Gamma \vdash e_1.m(e_2) : \{\bar{r}\}\ \texttt{with}\ \{\bar{r}, m\} \cup \varepsilon_1 \cup \varepsilon_2}\ (\varepsilon\text{-MethCallResource})$$

$$\frac{\Gamma \vdash e_1 : \{\bar{\sigma}\}\ \texttt{with}\ \varepsilon_1 \quad \Gamma \vdash e_2 : \tau_2\ \texttt{with}\ \varepsilon_2 \quad \sigma_i := \texttt{def}\ m_i(y : \tau_2) : \tau\ \texttt{with}\ \varepsilon}{\Gamma \vdash e_1.m_i(e_2) : \tau\ \texttt{with}\ \varepsilon_1 \cup \varepsilon_2 \cup \varepsilon}\ (\varepsilon\text{-MethCallObj})$$

Notes:

- The ε judgements are to be applied to portions of the program where the methods are explicitly annotated with their effects.
- The rules ε-Var, ε-Resource, and ε-NewObj have in their antecedents an expression typed with no effect. Merely having an object or resource is not an effect; you must do something with it, like a call a method on it, in order for your program to have effects.
- ε-ValidImpl says that the return type and effects of the body of a method must agree with what its signature says.
- According to ε-MethCallResource, we can call any method on a resource. Doing so returns that same resource.

3 Capture Rules (Orange)

$$\frac{\varepsilon = effects(\Gamma') \quad \Gamma' \subseteq \Gamma \quad \Gamma', x : \{\bar{d}\ \texttt{captures}\ \varepsilon\} \vdash \overline{d = e}\ \texttt{OK}}{\Gamma \vdash\ \texttt{new}\ x \Rightarrow \overline{d = e} : \{x \Rightarrow \bar{d}\ \texttt{captures}\ \varepsilon\}}\ (\text{C-NewObj})$$

$$\frac{\Gamma \vdash e_1 : \{\bar{d}\ \texttt{captures}\ \varepsilon\}\ \texttt{with}\ \varepsilon_1 \quad \Gamma \vdash e_2 : \tau_2\ \texttt{with}\ \varepsilon_2 \quad d_i :=\ \texttt{def}\ m_i(y : \tau_2) : \tau}{\Gamma \vdash e_1.m_i(e_2) : \tau\ \texttt{with}\ \varepsilon_1 \cup \varepsilon_2 \cup effects(\tau_2)}\ (\text{C-MethCall})$$

- The capture judgements are to be applied when the program is not explicitly annotated with their effects. These rules perform a conservative effect analysis.
- The rule C-NewObj takes unnannotated methods and labels them using the `capture` keyword. Whereas d `with` ε means that execution of the method defined by d has the effect ε, d `captures` ε means that d has the authority to perform the effect ε, though it may not actually do so. We can think of `captures` as an upper bound on the effects of a program, while `with` is a tight upper bound on the effects of a program.
- C-MethCall performs a conservative effect analysis by concluding the effects of an expression to be those effects which it captures.

3.1 Definition of effects function

The *effects* function returns the set of effects of an expression as determined by our calculus thus far in a certain typing context. It recurses on sub-expressions, looking for effect annotations.

- $effects(\cdot) = \varnothing$
- $effects(\{\bar{r}\}) = \{(r, m) \mid r \in \bar{r}, m \in M\}$
- $effects(\{\bar{d}\ \texttt{with}\ \varepsilon\}) = \varepsilon$
- $effects(\{\bar{d}\ \texttt{captures}\ \varepsilon\}) = \varepsilon$
- $effects(\{\bar{\sigma}\}) = \bigcup_{\sigma \in \bar{\sigma}}\ effects(\sigma)$
- $effects(d\ \texttt{with}\ \varepsilon) = \varepsilon$