Framing Rules

Henry Blanchette

1 Definitions

Note: in this document "formula" refers to "precise formula," however gradual formulas will eventually be supported.

A **permission** is to either access a field, written $\mathsf{access}(e.f)$, or to assume a predicate holds of its arguments, written $\mathsf{assume}(\alpha_C(\overline{e}))$.

A formula ϕ **requires** a permission π if ϕ contains an access or assumption that π premits. The set of all permissions that ϕ requires (the set of permissions required to frame ϕ) is called the **requirements** of ϕ.

A formula ϕ **grants** permission π if it contains an adjuct that yields π.

A set of permissions Π **frames** a formula ϕ if and only if ϕ requires only permissions contained in Π, written

$$\Pi \vDash_I \phi.$$

The **footprint** of a formula ϕ is the smallest permission mask that frames ϕ, written

$$\lfloor \phi \rfloor.$$

A formula ϕ is **self-framing** if and only if for any set of permissions ϕ, $\Pi \vDash_I \phi$, written

$$\vdash_{\mathsf{frm}I} \phi.$$

2 Framing Algorithm

The following algorithm decides $\Pi \vDash_I \phi$ for a given set of permissions Π and formula ϕ.

$$
\begin{array}{lll}
\Pi \vDash_I \phi \iff \mathsf{match}\ \phi\ \mathsf{with} & & \\
\quad \mid\ v & \mapsto & \top \\
\quad \mid\ x & \mapsto & \top \\
\quad \mid\ e_1 \oplus e_2 & \mapsto & \Pi \vDash_I e_1, e_2 \\
\quad \mid\ e_1 \odot e_2 & \mapsto & \Pi \vDash_I e_1, e_2 \\
\quad \mid\ e.o.f & \mapsto & \Pi \vDash_I e.o, o.f \\
\quad \mid\ o.f & \mapsto & \mathsf{access}(e.o) \in \Pi \\
\quad \mid\ \texttt{result}, \texttt{id}, \texttt{old}(id), \texttt{this} & \mapsto & \top \\
\quad \mid\ n, o, \texttt{null}, \texttt{true}, \texttt{false} & \mapsto & \top \\
\quad \mid\ \texttt{acc}(e.f) & \mapsto & \Pi \vDash_I e \\
\quad \mid\ \phi_1 \circledast \phi_2 & \mapsto & \Pi \cup \mathsf{granted}(\phi_1) \cup \mathsf{granted}(\phi_2) \vDash_I \phi_1, \phi_2 \\
\quad \mid\ \alpha_C(\overline{e}) & \mapsto & \Pi \vDash_I \overline{e} \\
\quad \mid\ \texttt{if}\ e\ \texttt{then}\ \phi_1\ \texttt{else}\ \phi_2 & \mapsto & \Pi \vDash_I e, \phi_1, \phi_2 \\
\quad \mid\ \texttt{unfolding}\ \alpha_C(\overline{e})\ \texttt{in}\ \phi & \mapsto & \mathsf{assume}(\alpha_C(\overline{e})) \in \Pi \wedge \Pi \vDash_I \overline{e}
\end{array}
$$

The following algorithm produces the set of permissions granted by a given formula ϕ.

$$
\begin{array}{lll}
\mathsf{granted}\ \phi := \mathsf{match}\ \phi\ \mathsf{with} & & \\
\quad \mid\ \texttt{acc}(e.f) & \mapsto & \{\mathsf{access}(e.f)\} \\
\quad \mid\ \alpha_C(\overline{e}) & \mapsto & \{\mathsf{assume}(\alpha_C(\overline{e}))\} \\
\quad \mid\ \phi_1 \circledast \phi_2 & \mapsto & \mathsf{granted}(\phi_1) \cup \mathsf{granted}(\phi_2) \\
\quad \mid\ \_ & \mapsto & \varnothing
\end{array}
$$

3 Self-Framing

The following algorithm decides $\vdash_{\mathsf{frm}I} \phi$ for a given formula ϕ.

$$\vdash_{\mathsf{frm}I} \phi \iff \varnothing \vDash_I \phi$$