

# Verifier WLP Definitions

Jenna Wise, Johannes Bader, Jonathan Aldrich, Joshua Sunshine

October 30, 2018

## 1 Weakest liberal precondition calculus definitions over self-framed non-gradual formulas

$$\text{WLP}(\text{skip}, \hat{\phi}) = \hat{\phi}$$

$$\text{WLP}(s_1; s_2, \hat{\phi}) = \text{WLP}(s_1, \text{WLP}(s_2, \hat{\phi}))$$

$$\text{WLP}(T \ x, \hat{\phi}) = \hat{\phi}[\text{defaultValue}(T)/x]$$

$$\text{WLP}(x := e, \hat{\phi}) = \max_{\Rightarrow} \left\{ \hat{\phi}' \mid \hat{\phi}' \Rightarrow \hat{\phi}[e/x] \quad \wedge \quad \hat{\phi}' \Rightarrow \text{acc}(e) \right\}$$

$$\text{WLP}(\text{if } (x \odot y) \{s_1\} \text{ else } \{s_2\}, \hat{\phi}) =$$

$$\text{WLP}(x.f := y, \hat{\phi}) = \text{acc}(x.f) * \max_{\Rightarrow} \left\{ \hat{\phi}' \mid \hat{\phi}' * \text{acc}(x.f) * (x.f = y) \Rightarrow \hat{\phi} \quad \wedge \quad \hat{\phi}' * \text{acc}(x.f) \in \text{SATFORMULA} \right\}$$

$$\text{WLP}(x := \text{new } C, \hat{\phi}) = \max_{\Rightarrow} \left\{ \hat{\phi}' \mid \hat{\phi}' * (x \neq \text{null}) * \overline{\text{acc}(x.f_i) * (x.f_i = \text{defaultValue}(T_i))} \Rightarrow \hat{\phi} \right\}$$

where  $\text{fields}(C) = \overline{T_i f_i}$

$$\text{WLP}(y := z.m(\bar{x}), \hat{\phi}) = \text{undefined}$$

$$\text{WLP}(y := z.m_C(\bar{x}), \hat{\phi}) = \max_{\Rightarrow} \left\{ \hat{\phi}' \mid y \notin \text{FV}(\hat{\phi}') \quad \wedge \quad \hat{\phi}' \Rightarrow (z \neq \text{null}) * \text{pre}(C, m) \left[ z/\text{this}, \overline{x_i/\text{params}(C, m)_i} \right] \right.$$

$$\left. \wedge \quad \hat{\phi}' * \text{post}(C, m) \left[ z/\text{this}, \overline{x_i/\text{old}(\text{params}(C, m)_i)}, y/\text{result} \right] \Rightarrow \hat{\phi} \right\}$$

$$\text{WLP}(\text{assert } \phi_a, \hat{\phi}) = \max_{\Rightarrow} \left\{ \hat{\phi}' \mid \hat{\phi}' \Rightarrow \hat{\phi} \quad \wedge \quad \hat{\phi}' \Rightarrow \phi_a \right\}$$

$$\text{WLP}(\text{release } \phi_a, \hat{\phi}) =$$

$$\text{WLP}(\text{hold } \phi_a \{s\}, \widehat{\phi}) =$$

**Note:**

**Dynamic method calls.** Dynamic method calls are left undefined, because we are not verifying programs with dynamic dispatch at this time (all method calls should be static method calls). They are included in the grammar for future implementation.

**If & Release & hold.** Definitions coming soon.

**Predicates in the logic.** Although the grammar allows for abstract predicate families, we do not support them yet. Therefore, we assume formulas look like:

$$\phi ::= \text{true} \mid e \odot e \mid \text{acc}(e.f) \mid \phi * \phi$$

## 2 Helpful function definitions

TBD

## 3 Algorithmic WLP calculus definitions over self-framed non-gradual formulas