Imprecision Separation for Gradual Program Verification with Implicit Dynamic Frames

Kaige Liu

April 6, 2019

1 Syntax

$\widetilde{\phi} ::= \widehat{\phi} \mid ?_A * \phi \mid \widehat{\phi} * ?_C \mid ?_A * \phi * ?_C$

2 Classical and Accessibility Parts

Definition 2.1. *We define ϕ_A as the accessibility parts, and ϕ_C as the classical parts of the formula ϕ, conjuated by $*$.*

Lemma 2.2. *Let $\phi, \phi' \in$ SatFormula. Then:*

$$\phi_C \Rightarrow \phi'_C \wedge \phi_A \Rightarrow \phi'_A \implies \phi \Rightarrow \phi'$$

Proof. Assume $\phi_C \Rightarrow \phi'_C$ and $\phi_A \Rightarrow \phi'_A$. Take any formula component ϕ'' separted by * from ϕ' .

- If $\phi'' = \mathrm{acc}(e.f)$, ϕ'' is part of ϕ'_A. Since $\phi \Rightarrow \phi_A$, $\phi_A \Rightarrow \phi'_A$ and $\phi'_A \Rightarrow \phi''$, then $\phi_A \Rightarrow \phi''$ by transitivity of implication.
- Otherwise, if ϕ'' is not an accessibility predicate, ϕ'' is part of ϕ'_C. Since $\phi \Rightarrow \phi_C$, $\phi_C \Rightarrow \phi'_C$, $\phi'_C \Rightarrow \phi''$, $\phi \Rightarrow \phi''$ by transitivity of implication. componenet

Therefore, each component ϕ'' of ϕ' is implied by ϕ. We can conclude $\phi \Rightarrow \phi'$. $\square$

3 Concretization

Definition 3.1. $\gamma : \widetilde{\text{FORMULA}} \to \mathbb{P}(\text{FORMULA})$ *is defined as:*

$$\gamma(\widehat{\phi}) = \{\widehat{\phi}\}$$

$$\gamma(?_A * \phi * ?_C) = \begin{cases} \{\widehat{\phi}' \in \text{SATFORMULA} \mid \widehat{\phi}' \Rightarrow \phi\} & (\phi \in \text{SATFORMULA}) \\ undefined & otherwise \end{cases}$$

$$\gamma(?_A * \phi) = \begin{cases} \{\widehat{\phi}' \in \text{SATFORMULA} \mid \widehat{\phi}'_C \Leftrightarrow \phi_C \wedge \widehat{\phi}'_A \Rightarrow \phi_A\} & (\phi \in \text{SATFORMULA}) \\ undefined & otherwise \end{cases}$$

$$\gamma(\widehat{\phi} * ?_C) = \begin{cases} \{\widehat{\phi}' \in \text{SATFORMULA} \mid \widehat{\phi}'_A \Leftrightarrow \phi_A \wedge \widehat{\phi}'_C \Rightarrow \phi_C\} & (\widehat{\phi} \in \text{SATFORMULA}) \\ undefined & otherwise \end{cases}$$

4 Foorprint Formula

Definition 4.1. *Define a partial order on fields* $\leq$: FIELD $\times$ FIELD *as* $\langle e, f\rangle \leq \langle e', f'\rangle$ *iff* e' *contains* $e.f$ *in* e' *or* $(e = e') \wedge (f = f')$.
Lemma 4.2. $\leq$ *defines a partial order on* FIELD.

Proof. Proof for Lemma 4.2

- Reflexive: $\langle e, f\rangle \leq \langle e, f\rangle$ since $e = e$ and $f = f$ by definition.
- Anti-symmetric: Given $\langle e', f'\rangle \leq \langle e, f\rangle$ and $\langle e, f\rangle \leq \langle e', f'\rangle$. Suppose $e \neq e'$ or $f \neq f'$. Then if $e = e'$ and $f = f'$, we are done with the proof. Otherwise, assume $e \neq e' \vee f \neq f'$, then e contains $e'.f'$ and e' contains $e.f$, which implies $econtainse.f$. Contradiction.
- Transitive: Given $\langle e, f\rangle \leq \langle e', f'\rangle$ and $\langle e', f'\rangle \leq \langle e'', f''\rangle$. If $e' = e''$ and $f' = f''$, then clearly $\langle e, f\rangle \leq \langle e'', f''\rangle$. Otherwise, e' contains $e''.f''$. Also, e contains $e'.f'$. Therefore, e contains $e''.f''$.

□

Definition 4.3. *Define* $\mu(\cdot)$: STATFOOTPRINT $\to$ FORMULA. μ *maps a static footprint* A *to a formula* $\phi' = \mu(A)$. *For each* $\langle e, f\rangle \in A$,$\phi'$ *should contain* $acc(e.f)$. *Then,* ϕ' *is sorted in ascending order with the partial order* $\leq$.

The following lemma describe a property of the minimum foorprint of a formula, that A must contain all subchains of any chain of fields $e.f_1.f_2...f_n.f$.

Lemma 4.4. *Let* $\phi \in$ FORMULA, *and* $A = \min_{\subseteq}\{A' \in \text{STATFPRINT} | A' \vdash_{frm} \phi\}$. *Then, if* $\langle e.f_1.f_2...f_n, f\rangle \in A$, *then* $\forall 1 \leqslant m \leqslant n$, $\langle e.f_1...f_m - 1, f_m\rangle \in A$.

Proof. Proof for Lemma 4.4
Supppose for the sake of contradiction, that for some field access $e.f_1.f_2...f_n, f \in A$, there exists a maximum m, such that $e.f_1.f_2...f_{m-1}, f_m \notin A$. Then, since FFIELD is the only rule that concludes $A \vdash_{frm} e.f$, we can't conclude $A \vdash_{frm} e.f_1.f_2...f_{m-1}.f_m$. Since m is the maximum such that $e.f_1.f_2...f_{m-1}, f_m \notin A$, $e.f_1.f_2...f_m, f_{m+1} \in A$. Since FFIELD is the only rule that uses the fact $\langle e.f_1.f_2...f_m, f_{m+1}\rangle \in A$, but we can't conclude the $A \vdash_{frm} e.f_1.f_2...f_{m-1}.f_m$, removing $\langle e.f_1.f_2...f_m, f_{m+1}\rangle$ from A doesn't change any framing preperty of A. Contradicts that A is minimal. □

The following lemma claims the existence of a self-framed formula with a postfix of any given formula.

Lemma 4.5. *Let* $\phi \in$ FORMULA*, and* $A = \min_{\subseteq}\{A' \in \text{STATFPRINT} | A' \vdash_{frm} \phi\}$*. Then,* $\mu(A) * \phi$ *is self-framed.*

Proof. Proof for Lemma 4.5 First, we prove that $\emptyset \vdash_{frm} \mu(A)$. We proceed by induction on the first k accesibility predicates separated by $*$ of $\mu(A)$.

- **Base case**: k = 0. Clearly, $\emptyset \vdash_{frm}$ true.
- **Inductive step**: Let the first k accessibility predicates of $\mu(A)$, separated by $*$, be $\mu(A)_{1-k}$. Suppose for the first k accessibility predicates of $\mu(A)$, $\emptyset \vdash_{frm} \mu(A)_{[k]}$ (Induction Hypothesis). Let the $k+1$th accessibility predicate be $acc(e.f)$. Need to show $\emptyset \vdash_{frm} mu(A)_{[k+1]}$. We proceed by a sub-induction:
  Sub-Claim: Let $e = e'.f_1.f_2...f_m$**. Then,** $\forall 0 \leqslant n \leqslant m$**,** $\lfloor \mu(A)_{[k]} \rfloor \vdash_{frm} e'.f_1.f_2...f_n$**.**
  - **Sub-Base case**: n = 0. Since $\emptyset \vdash_{frm} e'$ for arbitrary non-field access, $\lfloor \mu(A)_{[k]} \rfloor \vdash_{frm} e'$.
  - **Sub-Inductive case**: For simplification, let $e'' = e'.f_1...f_n$. Suppose for some n, $\lfloor \mu(A)_{[k]} \rfloor \vdash_{frm} e'.f_1.f_2...f_n = e''$. We need to show $\lfloor \mu(A)_{[k]} \rfloor \vdash_{frm} e'.f_1.f_2...f_{n+1} = e''.f_{n+1}$. Since $A \vdash_{frm} \phi$ and A is minimal, by Lemma 4.4, $\langle e'', f_{n+1}\rangle \in A$. By definition of μ (Definition 4.3), $acc(e''.f_{n+1})$ appears in $\mu(A)$. Since $\mu(A)$ is sorted by partial order $\leq$, and e contains $e''.f_{n+1}$, therefore, $acc(e''.f_{n+1})$ must appear before the kth term of $\mu(A)$, so $acc(e''.f_{n+1})$ is in $\mu(A)_{[k]}$. By FFIELD, since $\langle e'', f_{n+1}\rangle \in \lfloor \mu(A)_{[k]} \rfloor$ and $\lfloor \mu(A)_{[k]} \rfloor \vdash_{frm} e''$ by induction hypothesis, $\lfloor \mu(A)_{[k]} \rfloor \vdash_{frm} e''.f_{n+1}$

  Therefore, $\lfloor \mu(A)_{[k]} \rfloor \vdash_{frm} e$. By SFACC, $A \vdash_{frm} acc(e.f)$. By SFSEPOP, $\emptyset \vdash_{frm} \mu(A)_{[k]}$ and $\lfloor \mu(A)_{[k]} \rfloor \vdash_{frm} acc(e.f)$, $\emptyset \vdash_{frm} \mu(A)_{[k]} * acc(e.f) = \mu(A)_{k+1}$

Therefore, by the induction, we proved that $\emptyset \vdash_{frm} \mu(A)$. By definition of μ, $\lfloor \mu(A) \rfloor = A$. $A \vdash_{frm} \phi$, so $\lfloor \mu(A) \rfloor \vdash_{frm} \phi$. By SFSEPOP, $\vdash_{frm} \mu(A)$, $\lfloor \mu(A) \rfloor \vdash_{frm} \phi$, we can finally conclude that $\emptyset \vdash_{frm} \mu(A) * \phi$. □

5 Frame Implication

Lemma 5.1. *Let* $\widehat{\phi}, \phi' \in$ FORMULA *such that* $\widehat{\phi} \Rightarrow \phi'$*, and* $A = \min_{\subseteq}\{A' \in \text{STATFOOTPRINT} \mid A' \vdash_{frm} \phi'\}$*. Then,* $A \subseteq \lfloor \widehat{\phi} \rfloor$

Proof. Let $\langle e, f\rangle \in A$ be arbitrary. Since A is the minimum footprint that frames phi', $\exists \phi'_0$ as a component of ϕ' separated by $*$ such that ϕ'_0 contains $e.f$. Let $E = \{\langle H, \rho, A\rangle \in \textsc{env} \mid \rho \models \widehat{\phi}\}$ and $E' = \{\langle H', \rho', A'\rangle \in \textsc{env} \mid \rho \models \phi'\}$. By definition of implication, since $\widehat{\phi} \Rightarrow \phi'$, $E \subseteq E'$. By the grammar of the verification language, ϕ'_0 can be either $e.f \odot k$ or $k \odot e.f$, in either case some possible value of $e.f$ is eliminated. $\exists o$ such that $\forall \langle H', \rho', A'\rangle \in E'$, $H, \rho \vdash e.f \Downarrow v$ and $v \neq o$. Since $E \subseteq E'$, $\forall \langle H, \rho, A\rangle \in E$, $H, \rho \vdash e.f \Downarrow v$ and $v \neq o$. Therefore, $\widehat{\phi}$ must contain a component ϕ_0 separated by $*$, such that ϕ_0 contains $e.f$. Since $\widehat{\phi}$ is self-framed, $\widehat{\phi}$ contains $\mathrm{acc}(e.f)$, and therefore $\langle e, f\rangle \in \lfloor \widehat{\phi} \rfloor$.

□

6 Consistent Formula Lifting

Definition 6.1. *Let* $\cdot \mathrel{\widetilde{\models}} \cdot \subseteq \mathrm{MEM} \times \widetilde{\textsc{Formula}}$ *be defined inductively as:*

$$\frac{m \models \widehat{\phi}}{m \mathrel{\widetilde{\models}} \widehat{\phi}}\ \widetilde{\textsc{EvalStatic}}$$

$$\frac{m \models \phi \qquad A \vdash_{frm} \phi \qquad \forall \langle e, f\rangle \in A. m \models acc(e.f)}{m \mathrel{\widetilde{\models}} ?_A * \phi * ?_C}\ \textsc{EvalGrad}$$

$$\frac{m \models \phi \qquad A \vdash_{frm} \phi \qquad \forall \langle e, f\rangle \in A. m \models acc(e.f)}{m \mathrel{\widetilde{\models}} ?_A * \phi}\ \textsc{EvalGrad}_A$$

$$\frac{m \models \phi}{m \mathrel{\widetilde{\models}} \phi * ?_C}\ \textsc{EvalGrad}_C$$

Lemma 6.2. *Consistent Formula Evaluation:*

$$\exists \widehat{\phi} \in \gamma(\widetilde{\phi}). m \models \widehat{\phi} \iff m \mathrel{\widetilde{\models}} \widetilde{\phi}$$

Proof. (Proof for Lemma 6.2)

Case $\widetilde{\phi} = \phi'$

It follows from definition of γ(3.1) and static that $\gamma(\widehat{\phi}) = \{\phi\}$ and that $static(\widehat{\phi}) = \phi$. Also, $m \models static(\widehat{\phi})$ if and only if $m \mathrel{\widetilde{\models}} \widehat{\phi}$(6). Therefore, the theorem trivially holds.

Case $\widetilde{\phi} = ?_A * \phi$

Applying the definitions of γ and $\widetilde{\models}$, the goal becomes:

$\exists \widehat{\phi}' \in \textsc{SatFormula}.(\widehat{\phi}'_c \Leftrightarrow \phi_c) \wedge (\widehat{\phi}'_A \Rightarrow \phi_A) \wedge (m \models \widehat{\phi}') \iff m \models \phi, \exists A \vdash_{frm} \phi. \forall \langle e, f \rangle \in A. m \models \mathrm{acc}(e.f)$

- Case $\Rightarrow$: Since $\widehat{\phi}'_C \Rightarrow \phi_C$ and $\widehat{\phi}'_A \Rightarrow \phi_A$, we can conclude $\widehat{\phi}' \Rightarrow \phi$ (by Lemma 2.2). Since $m \models \widehat{\phi}'$, by definition of implication, $m \models \phi$.
  Let $A = \min_{\subseteq}\{A' \in \textsc{StatFootprint} \mid A' \vdash_{frm} \phi\}$. $m \models \widehat{\phi}'$ implies that $\forall \langle e, f \rangle \in \lfloor \widehat{\phi}' \rfloor, m \models acc(e.f)$. We also have $\widehat{\phi}' \Rightarrow \phi$, so $A \subseteq \lfloor \widehat{\phi}' \rfloor$ by Lemma 5.1. Therefore, $\forall \langle e, f \rangle \in A, m \models acc(e.f)$
- Case $\Leftarrow$ Let $\widehat{\phi}' = \mu(A) * \phi$. Since we didn't add anything to the classical part of $\widehat{\phi}'$, $\widehat{\phi}'_C \Leftrightarrow \phi_C$ trivially holds. Since we only add more items to the accessibility part of the formula, $\widehat{\phi}'_A \Rightarrow \phi_A$ holds. Finally, since $\forall \langle e, f \rangle \in A. m \models \mathrm{acc}(e.f)$, by Definition 4.3, we know that $m \models \mu(A)$. Since $m \models \phi$, therefore, $m \models \mu(A) * \phi = \widehat{\phi}'$.

Case $\widetilde{\phi} = \widehat{\phi} * ?_C$

Applying the definitions of γ, the goal becomes:

$$\exists \widehat{\phi}' \in \textsc{SatFormula}. \widehat{\phi}'_A \Leftrightarrow \widehat{\phi}_A \wedge \widehat{\phi}'_C \Rightarrow \widehat{\phi}_C \wedge m \models \widehat{\phi}' \iff m \models \widehat{\phi}$$

- Case $\Rightarrow$: Since $\widehat{\phi}'_C \Rightarrow \widehat{\phi}_C$ and $\widehat{\phi}'_A \Rightarrow \widehat{\phi}_A$, $\widehat{\phi}' \Rightarrow \widehat{\phi}$. Also, since $m \models \widehat{\phi}'$, by definition of implication, $m \models \widehat{\phi}$.
- Case $\Leftarrow$ substitute $\widehat{\phi}$ for $\widehat{\phi}'$. All the 3 clauses on the left hand side trivially hold.

Case $\widetilde{\phi} = ?_A * \phi * ?_C$

Applying the definitions of γ, the goal becomes:

$$\exists \widehat{\phi}' \in \textsc{SatFormula}. \widehat{\phi}' \Rightarrow \phi \wedge m \models \widehat{\phi}' \iff m \models \phi, A \vdash_{frm} \phi, \forall \langle e, f \rangle \in A. m \models \mathrm{acc}(e.f)$$

- Case $\Rightarrow$: Since $\widehat{\phi}' \Rightarrow \phi$ and $m \models \widehat{\phi}'$, by definition of implication, $m \models \phi$.
  Let $A = \min_{\Rightarrow}\{A' \in \textsc{StatFootprint} \mid A' \vdash_{frm} \phi\}$. $m \models \widehat{\phi}'$ implies that $\forall \langle e, f \rangle \in \lfloor \widehat{\phi}' \rfloor, m \models acc(e.f)$. We also have $\widehat{\phi}' \Rightarrow \phi$, so $A \subseteq \lfloor \widehat{\phi}' \rfloor$ by Lemma 5.1. Therefore, $\forall \langle e, f \rangle \in A, m \models acc(e.f)$
- Case $\Leftarrow$: Let $\widehat{\phi}' = \mu(A) * \phi$. Since we didn't add anything to the classical part of $\widehat{\phi}'$, $\widehat{\phi}'_C \Rightarrow \phi_C$ trivially holds. Since we only add more items to the accessibility part of the formula, $\widehat{\phi}'_A \Rightarrow \phi_A$ holds. By Lemma 2.1, $\widehat{\phi}' \Rightarrow \phi$. Finally, since $\forall \langle e, f \rangle \in A. m \models \mathrm{acc}(e.f)$, by definition of μ (Definition 4.3), we know that $m \models \mu(A)$. Since $m \models \phi$, therefore, $m \models \mu(A) * \phi = \widehat{\phi}'$.

$\square$

7 Implication

Definition 7.1. *Let* $\cdot \mathrel{\widetilde{\Rightarrow}} \cdot \subseteq \widetilde{\textsc{Formula}} \times \widetilde{\textsc{Formula}}$ *be defined inductively as:*

$$\frac{\widehat{\phi} \in \textsc{SatFormula} \qquad \widehat{\phi} \Rightarrow \text{static}(\widetilde{\phi}')}{\widehat{\phi} \mathrel{\widetilde{\Rightarrow}} \widetilde{\phi}'}\ \textsc{ImplStatic}$$

$$\frac{\widehat{\phi} \in \textsc{SatFormula} \qquad \widehat{\phi} \Rightarrow \phi \qquad \widehat{\phi} \Rightarrow \text{static}(\widetilde{\phi}')}{?_A * \phi * ?_C \mathrel{\widetilde{\Rightarrow}} \widetilde{\phi}'}\ \textsc{ImplGrad}$$

$$\frac{\widehat{\phi} \in \textsc{SatFormula} \qquad \widehat{\phi}_C \Leftrightarrow \phi_C \qquad \widehat{\phi}_A \Rightarrow \phi_A \qquad \widehat{\phi} \Rightarrow \text{static}(\widetilde{\phi}')}{?_A * \phi \mathrel{\widetilde{\Rightarrow}} \widetilde{\phi}'}\ \textsc{ImplGrad}_{\textsc{A}}$$

$$\frac{\widehat{\phi} \in \textsc{SatFormula} \qquad \widehat{\phi}_A \Leftrightarrow \phi_A \qquad \widehat{\phi}_C \Rightarrow \phi_C \qquad \widehat{\phi} \Rightarrow \text{static}(\widetilde{\phi}')}{\phi * ?_C \mathrel{\widetilde{\Rightarrow}} \widetilde{\phi}'}\ \textsc{ImplGrad}_{\textsc{C}}$$

Lemma 7.2. *Gradual implication can be equivalently defined as the following:*
For $\widetilde{\phi}, \widetilde{\phi}' \in \widetilde{\textsc{Formula}}$,

$$\widetilde{\phi} \mathrel{\widetilde{\Rightarrow}} \widetilde{\phi}' \iff \exists \widehat{\phi}.\widehat{\phi} \in \gamma(\widetilde{\phi}) \wedge \widehat{\phi} \Rightarrow \text{static}(\widetilde{\phi}')$$

Proof. (Proof for Lemma 7.2)

- Case $\widetilde{\phi}' = \widehat{\phi}'$: By IMPLSTATIC (Definition 7.1), $\exists \widehat{\phi} \in \textsc{SatFormula}.\widehat{\phi} \Rightarrow \mathit{static}(\widehat{\phi}') \iff \widehat{\phi} \mathrel{\widetilde{\Rightarrow}} \widehat{\phi}'$. By definition of γ (Definition 3.1), $\widehat{\phi} \in \gamma(\widehat{\phi}) \iff \widehat{\phi} \Rightarrow \mathit{static}(\widehat{\phi}')$. Therefore, $\widetilde{\phi} \mathrel{\widetilde{\Rightarrow}} \widetilde{\phi}' \iff \exists \widehat{\phi}.\widehat{\phi} \in \gamma(\widetilde{\phi}) \wedge \widehat{\phi} \Rightarrow \mathit{static}(\widetilde{\phi}')$.
- Case $\widetilde{\phi}' = ?_A * \phi' * ?_C$: By IMPLGRAD (Definition 7.1), $\exists \widehat{\phi} \in \textsc{SatFormula}.\widehat{\phi} \Rightarrow \phi' \iff \widehat{\phi} \mathrel{\widetilde{\Rightarrow}} \widehat{\phi}'$. By definition of γ (Definition 3.1), $\widehat{\phi} \in \gamma(\widehat{\phi}) \iff \widehat{\phi} \Rightarrow \phi'$. Therefore, $\widetilde{\phi} \mathrel{\widetilde{\Rightarrow}} \widetilde{\phi}' \iff \exists \widehat{\phi}.\widehat{\phi} \in \gamma(\widehat{\phi}) \wedge \widehat{\phi} \Rightarrow \mathit{static}(\widetilde{\phi}')$.
- Case $\widetilde{\phi}' = ?_A * \phi'$: By IMPLGRAD$_A$ (Definition 7.1), $\exists \widehat{\phi} \in \textsc{SatFormula}.\widehat{\phi}_C \Leftrightarrow \widehat{\phi}'_C \wedge \widehat{\phi}_A \Rightarrow \widehat{\phi}'_A \iff \widehat{\phi} \mathrel{\widetilde{\Rightarrow}} \widehat{\phi}'$. By definition of γ (Definition 3.1), $\widehat{\phi} \in \gamma(\widehat{\phi}) \iff \widehat{\phi}_C \Leftrightarrow \widehat{\phi}'_C \wedge \widehat{\phi}_A \Rightarrow \widehat{\phi}'_A$. Therefore, $\widetilde{\phi} \mathrel{\widetilde{\Rightarrow}} \widetilde{\phi}' \iff \exists \widehat{\phi}.\widehat{\phi} \in \gamma(\widetilde{\phi}) \wedge \widehat{\phi} \Rightarrow \mathit{static}(\widetilde{\phi}')$.
- Case $\widetilde{\phi}' = \widehat{\phi}' * ?_C$: By IMPLGRAD$_C$ (Definition 7.1), $\exists \widehat{\phi} \in \textsc{SatFormula}.\widehat{\phi}_A \Leftrightarrow \widehat{\phi}'_A \wedge \widehat{\phi}_C \Rightarrow \widehat{\phi}'_C \iff \widehat{\phi} \mathrel{\widetilde{\Rightarrow}} \widehat{\phi}'$. By definition of γ (Definition 3.1), $\widehat{\phi} \in \gamma(\widehat{\phi}) \iff \widehat{\phi}_A \Leftrightarrow \widehat{\phi}'_A \wedge \widehat{\phi}_C \Rightarrow \widehat{\phi}'_C$. Therefore, $\widetilde{\phi} \mathrel{\widetilde{\Rightarrow}} \widetilde{\phi}' \iff \exists \widehat{\phi}.\widehat{\phi} \in \gamma(\widetilde{\phi}) \wedge \widehat{\phi} \Rightarrow \mathit{static}(\widetilde{\phi}')$.

$\square$

Lemma 7.3. *Consistent Implication Lifting.*

$$\widetilde{\phi} \mathrel{\widetilde{\Rightarrow}} \widetilde{\phi}' \iff \exists \phi \in \gamma(\widetilde{\phi}), \exists \phi' \in \gamma(\widetilde{\phi}').\phi \Rightarrow \phi'$$

Proof. (Proof for Lemma 7.3)

- Case $\Rightarrow$: Let $\phi' = \mathit{static}(\widetilde{\phi}')$. By Lemma 7.2, $\exists \phi \in \gamma(\widetilde{\phi}).\phi \Rightarrow \mathit{static}(\widetilde{\phi}')$, where $\mathit{static}(\widetilde{\phi}') = \phi'$.
- Case $\Leftarrow$: By definition of γ, in all 4 cases, $\forall \phi' \in \gamma(\widetilde{\phi}').\phi' \Rightarrow \mathit{static}(\widetilde{\phi}')$. Since $\phi \Rightarrow \phi'$, we know $\phi \Rightarrow \mathit{static}(\widetilde{\phi}')$ by transitivity of implication. Therefore, the right hand side implies:

$$\exists \phi \in \gamma(\widetilde{\phi}).\phi \Rightarrow \mathit{static}(\widetilde{\phi}')$$

. By Lemma 7.2, $\widetilde{\phi} \mathrel{\widetilde{\Rightarrow}} \widetilde{\phi}'$. Therefore, we can replace the inference rule by

$\square$

8 Modified sWLP

We redefine $\widetilde{\text{sWLP}}$ to adjust to the new definition of gradual formulas:

$$\widetilde{\text{sWLP}}^m(\bar{s}, \widetilde{\phi}) = \begin{cases} \widehat{\phi}'_n \cdot \widetilde{\phi}_{n-1} \cdot ... \cdot \widetilde{\phi}_1 \cdot \text{nil} & \widetilde{\phi}_p \text{ and } \widetilde{\phi}_n \text{ precise} \\ ?_A * \widehat{\phi}'_n \cdot \widetilde{\phi}_{n-1} \cdot ... \cdot \widetilde{\phi}_1 \cdot \text{nil} & \widetilde{\phi}_p \text{ or } \widetilde{\phi}_n \text{ acc imprecise } \wedge \widetilde{\phi}_p \text{ and } \widetilde{\phi}_n \text{ classically precise} \\ \widehat{\phi}'_n * ?_C \cdot \widetilde{\phi}_{n-1} \cdot ... \cdot \widetilde{\phi}_1 \cdot \text{nil} & \widetilde{\phi}_p \text{ and } \widetilde{\phi}_n \text{ acc precise } \wedge \widetilde{\phi}_p \text{ or } \widetilde{\phi}_n \text{ classically imprecise} \\ ?_A * \widehat{\phi}'_n * ?_C \cdot \widetilde{\phi}_{n-1} \cdot ... \cdot \widetilde{\phi}_1 \cdot \text{nil} & \widetilde{\phi}_p \text{ or } \widetilde{\phi}_n \text{ acc imprecise } \wedge \widetilde{\phi}_p \text{ or } \widetilde{\phi}_n \text{ classically imprecise} \end{cases}$$

$$\text{where } \widetilde{\phi}_n \cdot \widetilde{\phi}_{n-1} \cdot ... \cdot \widetilde{\phi}_1 \cdot \text{nil} = \widetilde{\text{WLP}}(\bar{s}, \widetilde{\phi})$$

$$\widehat{\phi}'_n = \begin{cases} \min_{\Rightarrow}\{\widehat{\phi}'_n \mid \mathit{static}(\widetilde{\phi}_n) \Rightarrow \widehat{\phi}'_n * \widetilde{\phi}'_p \wedge \widehat{\phi}'_n * \widetilde{\phi}_p \in \textsc{SatFormula}\} & \text{if } \widetilde{\phi}'_p \text{ acc precise} \\ \min_{\Rightarrow}\{\widehat{\phi}'_n \mid \mathit{static}(\widetilde{\phi}_n) \Rightarrow \widehat{\phi}'_n \wedge \lfloor \widehat{\phi}'_n \rfloor = \emptyset\} & \text{otherwise} \end{cases}$$

$$\text{and } \widetilde{\phi}'_p = \mathit{mpre}(m)[z, x/\texttt{this,mparam(m)}]$$

9 Dynamic Semantics with Residual Checks

Definition 9.1. *Naive dynanimc semantics of GVL.*
Let $\langle H, \langle \rho_n, A_n, s_n\rangle \cdot ... \cdot \langle \rho_1, A_1, s_1\rangle \cdot \mathit{nil}\rangle, \langle H, \langle \rho'_n, A'_n, s'_n\rangle \cdot ... \cdot \langle \rho'_1, A'_1, s'_1\rangle \cdot \mathit{nil}\rangle \in \textsc{State}$. *If*

$$\langle H, \langle \rho_n, A_n, s_n\rangle \cdot ... \cdot \langle \rho_1, A_1, s_1\rangle \cdot \mathit{nil}\rangle \longrightarrow \langle H, \langle \rho'_n, A'_n, s'_n\rangle \cdot ... \cdot \langle \rho'_1, A'_1, s'_1\rangle \cdot \mathit{nil}\rangle$$

holds, and

$$\bar{\phi} = \widetilde{\text{sWLP}}(s_n \cdot ... \cdot s_1, \mathit{true})$$

then

$$\langle H, \langle \rho_n, A_n, s_n\rangle \cdot ... \cdot \langle \rho_1, A_1, s_1\rangle\rangle \widetilde{\longrightarrow} \begin{cases} \langle H, \langle \rho'_n, A'_n, s'_n\rangle \cdot ... \cdot \langle \rho'_1, A'_1, s'_1\rangle\rangle & (\forall i \leq n \ \langle H, \rho'_i, A'_i\rangle \mathrel{\widetilde{\models}} \bar{\phi}_i) \\ \mathit{error} & (\mathit{otherwise}) \end{cases}$$

Definition 9.2. *Dynamic semantics with residual checks.*

$$\frac{\langle H, \langle \rho_n, A_n, (s; s_n)\rangle \cdot ...\rangle \longrightarrow \langle H, \langle \rho'_n, A'_n, s_n\rangle \cdot ...\rangle}{\langle H, \langle \rho_n, A_n, (s; s_n)\rangle \cdot ...\rangle \widetilde{\longrightarrow} \langle H, \langle \rho'_n, A'_n, s_n\rangle \cdot ...\rangle} \text{ SsL\tilde{o}cal}$$

$$method(m) = T_r \quad m(T x') \texttt{ requires } \widetilde{\phi}_p \texttt{ ensures } \widetilde{\phi}_q\{r\} \quad H, \rho \vdash z \Downarrow o \quad H, \rho \vdash x \Downarrow v$$
$$\rho' = [\texttt{this} \mapsto o, x' \mapsto v] \quad A' = \begin{cases} \lfloor \widetilde{\phi}_p \rfloor_{H,\rho'} & \text{if } \widetilde{\phi}_p \text{ acc precise} \\ A & \text{otherwise} \end{cases}$$

$$\frac{\langle H, \rho', A'\rangle \,\widetilde{\models}\, \widetilde{\phi}_p \qquad \langle H, \rho', A'\rangle \,\widetilde{\models}\, \widetilde{\text{diff}}(\widetilde{\text{WLP}}(r, \widetilde{\phi}_q), \widetilde{\phi}_p)}{\langle H, \langle \rho, A, (y = z.m(x); s)\rangle \cdot ...\rangle \widetilde{\longrightarrow} \langle H, \langle \rho', A', r\rangle \cdot \langle \rho, A \setminus A', (y = z.m(x); s)\rangle \cdot ...\rangle} \text{ SsC\tilde{a}ll}$$

$$mpost(m) = \widetilde{\phi}_q \quad \rho'' = \rho[y \mapsto \rho'(result)] \quad \widetilde{\phi}'_q = \widetilde{\phi}_q[z, x, y/this, old(mparam(m), result)]$$
$$\widetilde{\phi}'_p = mpre(m)[z, x/\texttt{this,mparam(m)}] \quad \widetilde{\phi} = \text{s}\widetilde{\text{WLP}}_n(s_n \cdot ..., true)$$

$$\frac{\langle H, \rho'', A \cup A'\rangle \,\widetilde{\models}\, \widetilde{\text{diff}}(\widetilde{\phi}, \widetilde{\text{SP}}(y := z.m(x), \widetilde{\phi}))}{\langle H, \langle \rho', A', \texttt{skip}\rangle \cdot \langle \rho, A \setminus A', (y = z.m(x); s_n)\rangle \cdot S\rangle \widetilde{\longrightarrow} \langle H, \langle \rho'', A \cup A', s_n\rangle\rangle \cdot S\rangle} \text{ SsCal\tilde{l}Finish}$$

Definition 9.3. *Strongest postconditions.*
Let SP : Stmt × Formula → Formula *be defined as:*

$$\text{SP}(s, \phi) = \min_{\Rightarrow}\{\phi' \in \text{Formula} \mid \phi \Rightarrow \text{WLP}(s, \phi')\}$$

Let $\widetilde{\text{SP}} : \text{Stmt} \times \widetilde{\text{Formula}} \to \widetilde{\text{Formula}}$ *be defined as the Consistent Function Lifting of* SP*:*

$$\widetilde{\text{SP}}(s, \widetilde{\phi}) = \alpha(\{\text{SP}(s, \phi) \mid \phi \in \gamma(\widetilde{\phi})\})$$

Definition 9.4. *Reducing formulas.*
Let diff : Formula × Formula → Formula *be defined as:*

$$\text{diff}(\phi_j, \phi_k) = \max_{\Rightarrow}\{\phi \in \text{Formula} \mid (\phi * \phi_k \Rightarrow \phi_j) \wedge (\phi * \phi_k \in \text{SatFormula})\}$$

(Note: $\phi * \phi_k$ is not well ordered, but it should not affect evaluation of formula or formula implication.)
Let $\widetilde{diff} : \widetilde{\text{Formula}} \times \widetilde{\text{Formula}} \to \widetilde{\text{Formula}}$ *be defined as:*

$$\widetilde{\text{diff}}(\widetilde{\phi}_j, \widetilde{\phi}_k) = \begin{cases} \text{diff}(\widetilde{\phi}_j, \text{static}(\widetilde{\phi}_k)) & (\widetilde{\phi}_j \textit{ precise}) \\ ?_A * \text{diff}(\phi_j, \text{static}(\widetilde{\phi}_k)) & (\widetilde{\phi}_j = ?_A * \phi_j) \\ \text{diff}(\phi_a, \text{static}(\widetilde{\phi}_j)) * ?_C & (\widetilde{\phi}_j = \phi_j * ?_C) \\ ?_A * \text{diff}(\phi_j, \text{static}(\widetilde{\phi}_k)) * ?_C & (\widetilde{\phi}_j = ?_A * \phi_j * ?_C) \end{cases}$$

Lemma 9.5. *Dynamic semantics with residual checks is equivalent with full checks.*

Proof. Proof for Lemma 9.5. We prove the lemma by cases on the state $\langle H, S\rangle$. We assume $\langle H, S\rangle$ is valid. We will show after one step of evaluation $\langle H, S\rangle \longrightarrow \langle H, S'\rangle$, the state $\langle H, S'\rangle$ satisfies residual checks if the state H, S' satisfies the residual checks. The other direction (residual checks satisfied if full checks satisfied) is trivial since residual check is a subset of the full check.

Case Ss$\tilde{\text{L}}$ocal

In this case, $S = \langle \rho_n, A_n, s; s_n\rangle \cdot ...$, where s does not involve a method call. After one step of evaluation, assume $\langle H, S\rangle \longrightarrow \langle H, S'\rangle$, where $S' = \langle \rho'_n, A'_n, s_n\rangle \cdot ....$ The naive semantics proposes to perform the following check:

$$\langle H, \rho', A'_n\rangle \mathrel{\widetilde{\models}} \widetilde{\phi}$$

where $\widetilde{\phi} = \mathrm{s}\widetilde{\mathrm{WLP}}_n(s_n \cdot ... \cdot s_1, true)$. By assumption, the state we leave must be valid:

$$\langle H, \rho_n, A_n\rangle \mathrel{\widetilde{\models}} \mathrm{s}\widetilde{\mathrm{WLP}}_n(s; s_n \cdot ... \cdot s_1 \cdot nil, true) = \widetilde{\mathrm{WLP}}(s, \widetilde{\phi})$$

By definition of $\widetilde{\mathrm{SP}}$,

$$\langle H, \rho'_n, A'_n\rangle \mathrel{\widetilde{\models}} \widetilde{\mathrm{SP}}(s, \widetilde{\mathrm{WLP}}(s, \widetilde{\phi}))$$

By definition of $\widetilde{SP}$, we also know that for arbitrary precise formula ϕ,

$$\widetilde{\mathrm{SP}}(s, \widetilde{\mathrm{WLP}}(s, \phi)) \Rightarrow \phi$$

Therefore, by setting $\phi = \mathrm{static}(\widetilde{\phi})$,

$$\begin{aligned}
\mathrm{static}(\widetilde{\mathrm{SP}}(s, \widetilde{\mathrm{WLP}}(s, \widetilde{\phi}))) &= \widetilde{\mathrm{SP}}(s, \mathrm{static}(\widetilde{\mathrm{WLP}}(s, \widetilde{\phi}))) \\
&= \widetilde{\mathrm{SP}}(s, \widetilde{\mathrm{WLP}}(s, \mathrm{static}(\widetilde{\phi}))) \\
&\Rightarrow \mathrm{static}(\widetilde{\phi})
\end{aligned}$$

Therefore, $\langle H, \rho', A'_n\rangle \mathrel{\widetilde{\models}} \mathrm{static}(\widetilde{\phi})$, and therefore $\langle H, \rho', A'_n\rangle \mathrel{\widetilde{\models}} \widetilde{\phi}$ by definition of consistent formula lifting.

Case Ss$\tilde{\text{C}}$all

In this case, $S = \langle \rho, A, (y = z.m(x); s)\rangle \cdot ....$ Recalling the definitions in SsCall,

$$m(x) \text{ requires } \widetilde{\phi}_p \text{ ensures } \widetilde{\phi}_q\{r\}$$

Let $\bar{\phi} = \widetilde{\text{sWLP}}(r \cdot (y = z.m(x); s) \cdot ..., \text{true})$. Let $\widetilde{\phi} = \bar{\phi}_{|\bar{\phi}|}$ and $\widetilde{\phi}' = \bar{\phi}_{|\bar{\phi}|-1}$.
After one step of evaluation, $\langle H, S\rangle \widetilde{\longrightarrow} \langle H, S'\rangle$, where

$$S' = \langle \rho', A', r)\rangle \cdot \langle \rho, A \setminus A', y = z.m(x); s\rangle \cdot ...$$

The naive semantics proposes to perform the following 3 checks:

1. $\langle H, \rho', A'\rangle \mathrel{\widetilde{\models}} \widetilde{\phi}_p$: according to $\widetilde{\text{SsCall}}$, this check is performed explicitly.

2. $\langle H, \rho', A'\rangle \mathrel{\widetilde{\models}} \widetilde{\phi} = \widetilde{\text{WLP}}(r, \widetilde{\phi}_q)$: by definition of $\widetilde{\text{diff}}$, since we checked for $\langle H, \rho', A'\rangle \mathrel{\widetilde{\models}} \widetilde{\phi}_p$ explicitly, the naive check is reduced to:

$$\langle H, \rho', A'\rangle \mathrel{\widetilde{\models}} \widetilde{\text{diff}}(\widetilde{\text{WLP}}(r, \widetilde{\phi}_q), \widetilde{\phi}_p)$$

   which is checked explicitly in the residual checks.

3. $\langle H, \rho, A \setminus A'\rangle \mathrel{\widetilde{\models}} \widetilde{\phi}'$: let $\widetilde{\phi}_n = \widetilde{\text{sWLP}}_n((y = z.m(x); s) \cdot ..., true)$.

   - Case $\widetilde{\phi}'_p$ acc precise: by definition of $\widetilde{\text{sWLP}}^m$, $static(\widetilde{\phi}') = \widehat{\phi}'$, where

$$\widehat{\phi}' = min_{\Rightarrow}\{\widehat{\phi}'' \mid static(\widetilde{\phi}_n) \Rightarrow \widehat{\phi}'' * \widetilde{\phi}'_p \wedge \widehat{\phi}'' * \widetilde{\phi}'_p \in \text{SatFormula}\}$$

     By the dynamic semantics, we can assume the state we left must passed the check of the naive semantics, therefore $\langle H, \rho, A\rangle \mathrel{\widetilde{\models}} \widetilde{\phi}_n$. Since $\widetilde{\phi}_n \Rightarrow \widehat{\phi}'$, we know that $\langle H, \rho, A\rangle \mathrel{\widetilde{\models}} \widehat{\phi}'$. We also know that $\lfloor \widehat{\phi}' \rfloor \cap \lfloor \widetilde{\phi}_p \rfloor = \emptyset$ since $\widehat{\phi}' * \widetilde{\phi}_p \in \text{SatFormula}$. Since $A' = \lfloor \widetilde{\phi}_p \rfloor$ by SsCall, we can conclude that $\langle H, \rho, A \setminus A'\rangle \mathrel{\widetilde{\models}} \widehat{\phi}'$, since removing each field in A' must not appear in $\widehat{\phi}'$. Therefore,

$$\langle H, \rho, A \setminus A'\rangle \mathrel{\widetilde{\models}} static(\widetilde{\phi}')$$

     so we can conclude

$$\langle H, \rho, A \setminus A'\rangle \mathrel{\widetilde{\models}} \widetilde{\phi}'$$

   - Case $\widetilde{\phi}'_p$ acc imprecise:by definition of $\widetilde{\text{sWLP}}^m$, $\widetilde{\phi}' = \widehat{\phi}'$, where

$$\widehat{\phi}' = min_{\Rightarrow}\{\widehat{\phi}'' \mid static(\widetilde{\phi}_n) \Rightarrow \widehat{\phi}'' \wedge \lfloor \widehat{\phi}'' \rfloor = \emptyset\}$$

     By the dynamic semantics, we can assume the state we left must passed the check of the naive semantics, therefore $\langle H, \rho, A\rangle \mathrel{\widetilde{\models}} \widetilde{\phi}_n$. Since $\widetilde{\phi}_n \Rightarrow \widehat{\phi}'$, we know that $\langle H, \rho, A\rangle \mathrel{\widetilde{\models}} \widehat{\phi}'$. We also know that $A' = A$, and therefore $A \setminus A' = \emptyset$. Since $\lfloor \widehat{\phi}' \rfloor = \emptyset$, we can conclude that

$$\langle H, \rho, A \setminus A'\rangle \mathrel{\widetilde{\models}} \widehat{\phi}'$$

     since the formula has empty footprint, and $A \setminus A'$ is also empty. Since $\widetilde{\phi}' = \widehat{\phi}'$, we finally conclude that

$$\langle H, \rho, A \setminus A'\rangle \mathrel{\widetilde{\models}} \widetilde{\phi}'$$

Case SsCall$\widetilde{\ }$Finish

In this case, $s = y := z.m(x)$ and $S = \langle \rho', A', \texttt{skip} \cdot \langle \rho, A, s; s_n \rangle \cdot ...$ The naive semantics proposes to perform the following check:

$$\langle H, \rho'', A \cup A' \rangle \mathrel{\widetilde{\models}} \widetilde{\phi}$$

where $\widetilde{\phi} = \widetilde{\text{sWLP}}_n(s_n \cdot ... \cdot s_1, true)$. By assumption, the state before the method call must be valid:

$$\langle H, \rho'', A \cup A' \rangle \mathrel{\widetilde{\models}} \widetilde{\text{sWLP}}_n(s; s_n \cdot ... \cdot s_1 \cdot nil, true) = \widetilde{\text{WLP}}(s, \widetilde{\phi})$$

By definition of $\widetilde{\text{SP}}$, since state $\langle H, \rho', A' \rangle$ is reached by execution of s,

$$\langle H, \rho', A' \rangle \mathrel{\widetilde{\models}} \widetilde{\text{SP}}(s, \widetilde{\text{WLP}}(s, \widetilde{\phi}))$$

Therefore, the check proposed in the naive semantics is reduced to

$$\langle H, \rho', A' \rangle \mathrel{\widetilde{\models}} \widetilde{\text{diff}}(\widetilde{\phi}, \widetilde{\text{SP}}(s, \widetilde{\text{WLP}}(s, \widetilde{\phi})))$$

□

10 Soundness

We now establish the soundness of GVL with full checks. Since in the previous section we proved the dynmaic semantics with residual checks is equivalent to full checks, we proceed the proof of soundness only with full checks, and therefore extensible to residual checks.

Definition 10.1. *We call the state $\langle H, \langle \rho_n, A_n, s_n \rangle \cdot ... \cdot \langle \rho_1, A_1, s_1 \rangle \cdot nil \rangle \in \textsc{State}$ valid if $\langle H, \rho_i, A_i \rangle \models sWLP_i(s_n \cdot ... \cdot s_1 \cdot nil, true)$ for all $1 \le i \le n$.*

Lemma 10.2. *(Progress)*
If $\langle H, S \rangle \in \textsc{State}$ is a valid state, then for some $\langle H', S' \rangle \in \textsc{State}$, $\langle H, S \rangle \widetilde{\longrightarrow} \langle H', S' \rangle$ or $\langle H, S \rangle \widetilde{\longrightarrow}$ ***error****.*

Proof. Proof for Progress (Lemma 10.2).
We are given a valid state $\langle H, S \rangle \in \textsc{State}$. Soundness of SVL implies that $\longrightarrow$ will step. By definition of the naive dynamic semantics, $\widetilde{\longrightarrow}$ either steps as in SVL if the checks succeed, or go to the error state. □

Lemma 10.3. *(Preservation)*
If $\langle H, S \rangle$ is a valid state and for some $\langle H', S' \rangle \in \textsc{State}$, $\langle H, S \rangle \widetilde{\longrightarrow} \langle H', S' \rangle$ then $\langle H', S' \rangle$ is a valid state.

Proof. Proof for Progress (Lemma 10.3).
We are given a valid state $\langle H, S \rangle \in \textsc{State}$. Soundness of SVL says that $\longrightarrow$ will step to another valid state.

- If the explicit check succeeds, By definition of the naive dynamic semantics, $\widetilde{\longrightarrow}$ either steps as in SVL if the checks succeed, or go to the error state.
- If the explicit check fails, then the program steps to error state, which trivially satisfies preservation.

$\square$

11 Static Gradual Gaurantee of GVL

Lemma 11.1. *Let* $p \in \textsc{Stmt}$ $\widetilde{\phi}, \widetilde{\phi}' \in \textsc{Formula}$ *such that* $\widetilde{\phi} \sqsubseteq \widetilde{\phi}'$*, then* $\widetilde{\mathrm{WLP}}(s, \widetilde{\phi}) \sqsubseteq \widetilde{\mathrm{WLP}}(s, \widetilde{\phi}')$*.*
Lemma 11.2. *Let* $\bar{s} \in ???$ $\widetilde{\phi}, \widetilde{\phi}' \in \textsc{Formula}$ *such that* $\widetilde{\phi} \sqsubseteq \widetilde{\phi}'$*, then* $\forall 1 \leq i \leq n, \widetilde{\mathrm{sWLP}}_i(s, \widetilde{\phi}) \sqsubseteq \widetilde{\mathrm{sWLP}}_i(s, \widetilde{\phi}')$*.*
Lemma 11.3. *Let* $p_1, p_2 \in \textsc{Program}$ *such that* $p_1 \sqsubseteq p_2$*, then if* p_1 *is valid, then* p_2 *is valid.*

Proof. Validity of functions and programs relies on $\widetilde{\mathrm{sWLP}}$. Therefore, we prove the lemma based on different cases in WLP. $\square$

12 Dynamic Gradual Guarantee of GVL

Definition 12.1. *(State Precision) Let* $\pi_1, \pi_2 \in \textsc{State}$*. Then* π_1 *is more precise than* π_2*, written* $\pi_1 \lesssim \pi_2$*, if and only if all of the following applies:*

1. π_1 *and* π_2 *have identical heap and stacks of size n.*
2. *The stack of variable environments and stack of statements is identical.*
3. *Let* $A^1_{1\ldots n}$ *and* $A^2_{1\ldots n}$ *be the stack of footprints of* π_1 *and* π_2*, respectively. Then, the following holds for* $1 \leq m \leq n$*:*
$$\bigcup_{i=m}^{n} A^1_i \subseteq \bigcup_{i=m}^{n} A^2_i$$

Lemma 12.2. *Let* $p_1, p_2 \in \textsc{Program}$ *such that* $p1 \sqsubseteq p2$*, and* $pi \in \textsc{State}$ *such that* $\pi_1 \lesssim \pi_2$*. If* $\pi_1 \widetilde{\rightarrow}_{p_1} \pi_1'$*, then* $\pi_2 \widetilde{\rightarrow}_{p_2} \pi_2'$ *for some* $\pi_2' \in \textsc{State}$ *and* $\pi_1' \lesssim \pi_2'$*.*

Proof. We analyze the definition of $\widetilde{\rightarrow}$. Increasing imprecision of contracts will increase the imprecision of $\widetilde{\mathrm{sWLP}}$ by Lemma 11.2 and hence increase the chances that that the non- error case applies. Hence, if $\pi_1 \widetilde{\Rightarrow}_{p_1} \pi_1'$, then $\pi_2 \widetilde{\Rightarrow}_{p_2} \pi_2'$. Now all we need to prove is $\pi_1' \lesssim \pi_2'$. π_1' and π_2' have the same heap, stack size, variable environments and stack of statements, since the programs are identical, and difference in contracts doesn't affect heap and stack, so the first 2 properties of $\lesssim$ are trivial. We finally prove the third property of $\lesssim$:

$$\bigcup_{i=m}^{n-1} A'^1_i \subseteq \bigcup_{i=m}^{n-1} A'^2_i$$

The dynamic footprints satisfies the requiresment. Let stack size before executing the next statement be n. We assume the state before executing the statement satisfies for $1 \leq m \leq n$:

$$\bigcup_{i=m}^{n} A_i^1 \subseteq \bigcup_{i=m}^{n} A_i^2$$

We perform the analysis based on the naive dynamic sementics. Since the naive dynamic semantics changes the dynamic footprint according to SVL, we only need to analyze the 3 cases in SVL that changes the dynamic footprint: SsAlloc, SsCall and SsCallFinish

- Case SsAlloc: the only change to the dynamic footprint is to add $\langle o, f_i \rangle$ to both A_n^1 and A_n^2. Therefore, the property still holds after the statement is executed.
- Case SsCall: recalling from SsCall, let $\widetilde{\phi}^1$ represent the precondition in p_1 and $\widetilde{\phi}^2$ represents the precondition in p_2. Consider the following cases:
  1. $\widetilde{\phi}^2$ acc precise. Then, since $\widetilde{\phi}^1 \sqsubseteq \widetilde{\phi}^2$, $\widetilde{\phi}^1$ must also be acc precise, which implies $\widetilde{\phi}_a^1 = \widetilde{\phi}_a^2$. Therefore, $\lfloor \widetilde{\phi}^1 \rfloor = \lfloor \widetilde{\phi}^2 \rfloor$. Therefore, $A_{n+1}'^1 = A_{n+1}'^2$, which implies
  $$A_{n+1}'^1 \subseteq A_{n+1}'^2$$
  .
  2. $\widetilde{\phi}^2$ acc imprecise, then by SsCall, $A_{n+1}'^2 = A_n^2$. Since we know that $A_n^1 \subseteq A_n^2$ by plugging $m = n$ into assumption, and $A_n'^1 \subseteq A_n^1$ by framing rule, we can conclude that
  $$A_{n+1}'^1 \subseteq A_{n+1}'^2$$
  .

  Since we know
  $$\bigcup_{i=m}^{n} A_i^1 \subseteq \bigcup_{i=m}^{n} A_i^2$$
  and clearly A_n^1 is partitioned into $A_n'^1$ and $A_{n+1}'^1$ and A_n^2 is partitioned into $A_n'^2$ and $A_{n+1}'^2$, so we can conclude that when $1 \leq m \leq n$
  $$\bigcup_{i=m}^{n+1} A_i'^1 = \bigcup_{i=m}^{n} A_i^1 \subseteq \bigcup_{i=m}^{n} A_i^2 = \bigcup_{i=m}^{n+1} A_i'^2$$
  When, $m = n + 1$,
  $$\bigcup_{i=m}^{n+1} A_i'^1 = A_{n+1}'^1 \subseteq A_{n+1}'^2 = \bigcup_{i=m}^{n+1} A_i'^2$$
  Therefore, $\forall 1 \leq m \leq n + 1$:
  $$\bigcup_{i=m}^{n+1} A_i'^1 \subseteq \bigcup_{i=m}^{n+1} A_i'^2$$

- Case SsCallFinish: By SsCallFinish, $A'^1_{n-1} = A^1_n \cup A^1_{n-1}$ and $A'^2_{n-1} = A^2_n \cup A^2_{n-1}$. Since we can assume that $\forall 1 \leq m \leq n$:

$$\bigcup_{i=m}^{n} A^1_i \subseteq \bigcup_{i=m}^{n} A^2_i$$

Therefore, $\forall 1 \leq m \leq n-1$:

$$\bigcup_{i=m}^{n-1} A'^1_i = \bigcup_{i=m}^{n} A^1_i \subseteq \bigcup_{i=m}^{n} A^2_i = \bigcup_{i=m}^{n-1} A'^2_i$$

□

Appendix

The following proof for the non IDF version of SsCallFinish no longer works:
Let s be the statement $y := z.m(x)$.

$$
\begin{aligned}
&\widetilde{\mathrm{SP}}(s, \widetilde{\mathrm{WLP}}(s, \widetilde{\phi}))\\
=&\alpha(\{\min_{\Rightarrow}\{\phi'' * \phi_q \mid (\phi \Rightarrow \phi'') \wedge y \notin FV(\phi'') \wedge \phi'' \Rightarrow \phi_p\} \mid \phi \in \gamma(\mathrm{WLP}(s, \widetilde{\phi})), \phi_p \in \gamma(\widetilde{\phi}'_p), \phi_q \in \gamma(\widetilde{\phi}'_q)\})\\
=&\alpha(\{\min_{\Rightarrow}\{\phi'' * \phi_q \mid (\phi \Rightarrow \phi'') \wedge y \notin FV(\phi'') \wedge \phi'' \Rightarrow \phi_p\} \mid y \notin FV(\phi) \wedge (\phi \mathrel{\widetilde{\Rightarrow}} \widetilde{\phi}'_p) \wedge (\phi * \widetilde{\phi}'_q \mathrel{\widetilde{\Rightarrow}} \widetilde{\phi}),\\
&\phi_p \in \gamma(\widetilde{\phi}'_p), \phi_q \in \gamma(\widetilde{\phi}'_q)\})\\
=&\alpha(\{\phi * \phi_q \mid y \notin FV(\phi) \wedge (\phi \Rightarrow \phi_p) \wedge (\phi * \widetilde{\phi}'_q \mathrel{\widetilde{\Rightarrow}} \widetilde{\phi}), \phi_p \in \gamma(\widetilde{\phi}'_p), \phi_q \in \gamma(\widetilde{\phi}'_q)\})\\
=&\alpha(\{\phi * static(\widetilde{\phi}'_q) \mid y \notin FV(\phi) \wedge (\phi \Rightarrow static(\widetilde{\phi}'_p)) \wedge (\phi * \widetilde{\phi}'_q \mathrel{\widetilde{\Rightarrow}} \widetilde{\phi})\})
\end{aligned}
$$