Finding sources of bugs: Sufficient Incorrectness Logic

Flavio Ascari

University of Pisa

Joint work with Roberto Bruni, Roberta Gori and Francesco Logozzo

Contribution

1. Sufficient Incorrectness Logic: a complete proof system to find the source of incorrectness.
   - Separation SIL: find the source of memory bugs.
2. Many program logic for correctness and incorrectness: taxonomy to compare and assess the capabilities of some of them.

Sufficient Incorrectness Logic

Regular commands

$$
\begin{aligned}
c ::= {} & \texttt{skip} \\
\mid {} & \texttt{x := a} \\
\mid {} & \texttt{b?} \\
\mid {} & \texttt{x := nondet()}
\end{aligned}
\qquad
\begin{aligned}
r ::= {} & c \\
\mid {} & r_1 ; r_2 \\
\mid {} & r_1 + r_2 \\
\mid {} & r^*
\end{aligned}
$$

Hoare logic and Incorrectness Logic

HL

Partial correctness

$$\{P\}\ \mathsf{r}\ \{Q\}$$

$$[\![\mathsf{r}]\!]P \subseteq Q$$

All states reachable from P are in Q. If Q has no errors, r is safe for P.

IL

Incorrectness

$$[P]\ \mathsf{r}\ [Q]$$

$$[\![\mathsf{r}]\!]P \supseteq Q$$

All states in Q are reachable from P. Any error in Q is a true alarm of r on some input in P.

Backward under-approximation

$$[\![\overleftarrow{\mathsf{r}}]\!]Q \supseteq P$$

As first-order formula:

$$[\![\overleftarrow{\mathsf{r}}]\!]Q \supseteq P \quad \equiv \quad \forall \sigma \in P \,.\, \exists \sigma' \in Q \,.\, \sigma' \in [\![\mathsf{r}]\!]\sigma$$

"All states in P have at least one execution leading to Q"

Why SIL?

Key differences with IL:

- Not all states in Q must be reachable
- If Q are all errors, from *any* state in P begins an erroneous trace

Why SIL?

Key differences with IL:

- Not all states in Q must be reachable
- If Q are all errors, from *any* state in P begins an erroneous trace

Manifest errors

An error Q is manifest if

$$\forall \sigma \,.\, \exists \sigma' \in [\![\mathsf{r}]\!]\sigma \,.\, \sigma' \models Q$$

This is just

$$[\![\overleftarrow{\mathsf{r}}]\!]Q \supseteq \mathsf{true}$$

A proof system for backward under-approximation

Core rules

$$\frac{}{\langle\!\langle [\![\overleftarrow{\mathsf{c}}]\!] Q \rangle\!\rangle\ \mathsf{c}\ \langle\!\langle Q \rangle\!\rangle}\ \langle\!\langle\text{atom}\rangle\!\rangle \qquad \frac{P \subseteq P' \quad \langle\!\langle P' \rangle\!\rangle\ \mathsf{r}\ \langle\!\langle Q' \rangle\!\rangle \quad Q' \subseteq Q}{\langle\!\langle P \rangle\!\rangle\ \mathsf{r}\ \langle\!\langle Q \rangle\!\rangle}\ \langle\!\langle\text{cons}\rangle\!\rangle$$

$$\frac{\langle\!\langle P_1 \rangle\!\rangle\ \mathsf{r}_1\ \langle\!\langle Q \rangle\!\rangle \quad \langle\!\langle P_2 \rangle\!\rangle\ \mathsf{r}_2\ \langle\!\langle Q \rangle\!\rangle}{\langle\!\langle P_1 \cup P_2 \rangle\!\rangle\ \mathsf{r}_1 + \mathsf{r}_2\ \langle\!\langle Q \rangle\!\rangle}\ \langle\!\langle\text{choice}\rangle\!\rangle \qquad \frac{\langle\!\langle P \rangle\!\rangle\ \mathsf{r}_1\ \langle\!\langle R \rangle\!\rangle \quad \langle\!\langle R \rangle\!\rangle\ \mathsf{r}_2\ \langle\!\langle Q \rangle\!\rangle}{\langle\!\langle P \rangle\!\rangle\ \mathsf{r}_1; \mathsf{r}_2\ \langle\!\langle Q \rangle\!\rangle}\ \langle\!\langle\text{seq}\rangle\!\rangle$$

$$\frac{\forall n \geq 0\,.\, \langle\!\langle Q_{n+1} \rangle\!\rangle\ \mathsf{r}\ \langle\!\langle Q_n \rangle\!\rangle}{\langle\!\langle \bigcup_{n \geq 0} Q_n \rangle\!\rangle\ \mathsf{r}^*\ \langle\!\langle Q_0 \rangle\!\rangle}\ \langle\!\langle\text{iter}\rangle\!\rangle$$

Additional rules

$$\frac{}{\langle\!\langle \emptyset \rangle\!\rangle\ \mathsf{r}\ \langle\!\langle Q \rangle\!\rangle}\ \langle\!\langle\text{empty}\rangle\!\rangle \qquad \frac{\langle\!\langle P_1 \rangle\!\rangle\ \mathsf{r}\ \langle\!\langle Q_1 \rangle\!\rangle \quad \langle\!\langle P_2 \rangle\!\rangle\ \mathsf{r}\ \langle\!\langle Q_2 \rangle\!\rangle}{\langle\!\langle P_1 \cup P_2 \rangle\!\rangle\ \mathsf{r}\ \langle\!\langle Q_1 \cup Q_2 \rangle\!\rangle}\ \langle\!\langle\text{disj}\rangle\!\rangle$$

$$\frac{}{\langle\!\langle Q \rangle\!\rangle\ \mathsf{r}^*\ \langle\!\langle Q \rangle\!\rangle}\ \langle\!\langle\text{iter0}\rangle\!\rangle \qquad \frac{\langle\!\langle P \rangle\!\rangle\ \mathsf{r}^*; \mathsf{r}\ \langle\!\langle Q \rangle\!\rangle}{\langle\!\langle P \rangle\!\rangle\ \mathsf{r}^*\ \langle\!\langle Q \rangle\!\rangle}\ \langle\!\langle\text{unroll}\rangle\!\rangle$$

$$\frac{\langle\!\langle P \rangle\!\rangle\ \mathsf{r}^*; \mathsf{r}\ \langle\!\langle Q_1 \rangle\!\rangle}{\langle\!\langle P \cup Q_2 \rangle\!\rangle\ \mathsf{r}^*\ \langle\!\langle Q_1 \cup Q_2 \rangle\!\rangle}\ \langle\!\langle\text{unroll-split}\rangle\!\rangle$$

Soundness and completeness

Validity of a triple: $[\![\overleftarrow{\mathsf{r}}]\!]Q \supseteq P$

Soundness

All provable triples (including additional rules) are valid.

Completeness

All valid triples are provable (using the core rules).

Backward analysis

The proof system favours backward analysis starting from the (error) postconditions.

Backward variant for atoms (assignment).

$$\frac{}{\langle\!\langle Q[a/x] \rangle\!\rangle \ \texttt{x := a} \ \langle\!\langle Q \rangle\!\rangle} \ \langle\!\langle \text{atom-a} \rangle\!\rangle$$

Backward analysis

The proof system favours backward analysis starting from the (error) postconditions.

Backward variant for atoms (guard).

$$\frac{}{\langle\!\langle Q \cap b \rangle\!\rangle \ \mathtt{b?} \ \langle\!\langle Q \rangle\!\rangle} \ \langle\!\langle \text{atom-g} \rangle\!\rangle$$

Backward analysis

The proof system favours backward analysis starting from the (error) postconditions.

Same postcondition for both branches.

$$\frac{\langle\!\langle P_1 \rangle\!\rangle\ \mathsf{r}_1\ \langle\!\langle Q \rangle\!\rangle \quad \langle\!\langle P_2 \rangle\!\rangle\ \mathsf{r}_2\ \langle\!\langle Q \rangle\!\rangle}{\langle\!\langle P_1 \cup P_2 \rangle\!\rangle\ \mathsf{r}_1 + \mathsf{r}_2\ \langle\!\langle Q \rangle\!\rangle}\ \langle\!\langle \text{choice} \rangle\!\rangle$$

Backward analysis

The proof system favours backward analysis starting from the (error) postconditions.

Backward iteration starting from final states Q_0.

$$\frac{\forall n \geq 0 \,.\, \langle\!\langle Q_{n+1} \rangle\!\rangle \ \mathsf{r} \ \langle\!\langle Q_n \rangle\!\rangle}{\langle\!\langle \bigcup_{n \geq 0} Q_n \rangle\!\rangle \ \mathsf{r}^* \ \langle\!\langle Q_0 \rangle\!\rangle} \ \langle\!\langle \text{iter} \rangle\!\rangle$$

Backward analysis

The proof system favours backward analysis starting from the (error) postconditions.

SIL can drop disjuncts going backward:

$$\frac{\langle\!\langle P \cup P' \rangle\!\rangle \ \mathsf{r} \ \langle\!\langle Q \rangle\!\rangle}{\langle\!\langle P \rangle\!\rangle \ \mathsf{r} \ \langle\!\langle Q \rangle\!\rangle} \ \langle\!\langle \text{cons}' \rangle\!\rangle$$

$$\frac{}{\langle\!\langle \emptyset \rangle\!\rangle \ \mathsf{r} \ \langle\!\langle Q \rangle\!\rangle} \ \langle\!\langle \text{empty} \rangle\!\rangle$$

Separation SIL

Separation SIL

Backward under-approximation + separation logic.
Pre and post are formulae in a logical language.

Separation SIL

Backward under-approximation + separation logic.
Pre and post are formulae in a logical language.

$$
\begin{aligned}
p, q \coloneqq\ & \mathsf{true} \mid \mathsf{false} \mid p \wedge q \mid p \vee q && \text{[propositional logic]} \\
\mid\ & \mathtt{a} = \mathtt{a} \mid \mathtt{a} \neq \mathtt{a} \mid \mathtt{a} \leq \mathtt{a} \mid \mathtt{a} \not\leq \mathtt{a} \mid \ldots && \text{[atomic propositions]} \\
\mid\ & \exists x.p && \text{[coherent logic]} \\
\mid\ & \mathbf{emp} \mid x \mapsto \mathtt{a} \mid x \not\mapsto \mid p * q && \text{[structural constructs]}
\end{aligned}
$$

Separation SIL example

Example program fragment (from Raad et al., 2020):

$$\mathsf{rclient} \triangleq x := [v];\ (\mathsf{r}_b + \texttt{skip})$$

$$\mathsf{r}_b \triangleq y := [v];\ \mathsf{free}(y);\ y := \mathsf{alloc}();\ [v] := y$$

Separation SIL example

Example program fragment (from Raad et al., 2020):

$$\mathsf{rclient} \triangleq x := [v];\ (\mathsf{r}_b + \texttt{skip})$$

$$\mathsf{r}_b \triangleq y := [v];\ \mathsf{free}(y);\ y := \mathsf{alloc}();\ [v] := y$$

ISL:

$$[v \mapsto z * z \mapsto -]\ \mathsf{rclient}\ [v \mapsto y * y \mapsto - * x \not\mapsto].$$

Separation SIL:

$$\langle\!\langle v \mapsto z * z \mapsto - * \mathsf{true} \rangle\!\rangle\ \mathsf{rclient}\ \langle\!\langle x \not\mapsto\ * \mathsf{true} \rangle\!\rangle$$

Separation SIL example (cont.d)

The SIL inference process is backward:

$$\langle\!\langle v \mapsto z * z \mapsto - * (x = z \vee x \not\mapsto) * \mathsf{true} \rangle\!\rangle$$

$$y := [v];$$

$$\langle\!\langle \underline{v \mapsto z * y = y'} * y' \mapsto - * (x = y' \vee x \not\mapsto) * \mathsf{true} \rangle\!\rangle$$

$$\langle\!\langle v \mapsto - * y \mapsto - * (x = y \vee x \not\mapsto) * \mathsf{true} \rangle\!\rangle$$

$$\mathsf{free}(y);$$

$$\langle\!\langle v \mapsto - * \underline{y \not\mapsto} * (x = y \vee x \not\mapsto) * \mathsf{true} \rangle\!\rangle$$

$$\langle\!\langle x \not\mapsto * v \mapsto - * \mathbf{emp} * \mathsf{true} \rangle\!\rangle$$

$$y := \mathsf{alloc}();$$

$$\langle\!\langle x \not\mapsto * v \mapsto - * \underline{y \mapsto y'} * \mathsf{true} \rangle\!\rangle$$

$$\langle\!\langle x \not\mapsto * v \mapsto - * \mathsf{true} \rangle\!\rangle$$

$$[v] := y$$

$$\langle\!\langle x \not\mapsto * \underline{v \mapsto y} * \mathsf{true} \rangle\!\rangle$$

$$\langle\!\langle x \not\mapsto * \mathsf{true} \rangle\!\rangle$$

Separation SIL: soundness and completeness

A separation SIL triple $\langle\!\langle p \rangle\!\rangle$ r $\langle\!\langle q \rangle\!\rangle$ is valid if

$$[\![\overleftarrow{\mathsf{r}}]\!] q \supseteq p$$

Separation SIL: soundness and completeness

A separation SIL triple $\langle\!\langle p \rangle\!\rangle$ r $\langle\!\langle q \rangle\!\rangle$ is valid if

$$[\![\overleftarrow{\mathsf{r}}]\!] q \supseteq p$$

Soundness

Any provable separation SIL triple is valid.

Relative Completeness

Suppose to have an oracle for implications. Then any valid separation SIL triple for a loop-free program is provable.

Separation SIL: completeness

Relative Completeness

Suppose to have an oracle for implications. Then any valid separation SIL triple for a loop-free program is provable.

- Expressiveness result on the assertion language
- Akin to Cook completeness for HL
- (Mostly) constructive proof

Taxonomy

Schema

Compare the available logics:

	Forward	Backward
Over	$[\![\mathsf{r}]\!]P \subseteq Q$ {HL}	$[\![\overleftarrow{\mathsf{r}}]\!]Q \subseteq P$ (NC)
Under	$[\![\mathsf{r}]\!]P \supseteq Q$ [IL]	$[\![\overleftarrow{\mathsf{r}}]\!]Q \supseteq P$ ⟪SIL⟫

$$\sigma \in [\![\overleftarrow{\mathsf{r}}]\!]\sigma' \iff \sigma' \in [\![\mathsf{r}]\!]\sigma$$

Consequence rule

Consequence rules follows the diagonal of the schema, so they suggest relations between HL-SIL and IL-NC.

Consequence rule

Consequence rules follows the diagonal of the schema, so they suggest relations between HL-SIL and IL-NC.
However, there aren't many.

HL-SIL: loosely related.

NC-IL: no relation.

r deterministic: (SIL) $\implies$ (HL) $\quad \forall \sigma' \in Q \,.\, \forall \sigma \in [\![\overleftarrow{\mathsf{r}}]\!] \sigma' \,.\, \sigma \in P$

r terminating: (HL) $\implies$ (SIL) $\quad \forall \sigma' \in Q \,.\, \exists \sigma \in [\![\overleftarrow{\mathsf{r}}]\!] \sigma' \,.\, \sigma \in P$

Approximation

Compare logics along the approximation axis.

HL-NC: strongly related.

IL-SIL: no relation.

$$\{P\}\ \mathsf{r}\ \{Q\} \iff (\neg P)\ \mathsf{r}\ (\neg Q)$$

Approximation

Compare logics along the approximation axis.

HL-NC: strongly related.

IL-SIL: no relation.

$$\{P\}\ \mathsf{r}\ \{Q\} \iff (\neg P)\ \mathsf{r}\ (\neg Q)$$

Difference between over- and under-approximation in the semantics of programs:

$$[\![\overleftarrow{\mathsf{r}}]\!][\![\mathsf{r}]\!]P \supseteq P \setminus \{\sigma \mid \sigma \text{ diverges}\}$$
$$[\![\mathsf{r}]\!][\![\overleftarrow{\mathsf{r}}]\!]Q \supseteq Q \setminus \{\sigma' \mid \sigma' \text{ unreachable}\}$$

Rule duality

Rule	SIL	IL	HL
atom	$\overline{\langle\!\langle [\![\overleftarrow{\mathsf{c}}]\!] Q \rangle\!\rangle\ \mathsf{c}\ \langle\!\langle Q \rangle\!\rangle}$	$\overline{[P]\ \mathsf{c}\ [[\![\mathsf{c}]\!] P]}$	$\overline{\{P\}\ \mathsf{c}\ \{[\![\mathsf{c}]\!] P\}}$
cons	$\dfrac{P \subseteq P' \quad \langle\!\langle P' \rangle\!\rangle\ \mathsf{r}\ \langle\!\langle Q' \rangle\!\rangle \quad Q' \subseteq Q}{\langle\!\langle P \rangle\!\rangle\ \mathsf{r}\ \langle\!\langle Q \rangle\!\rangle}$	$\dfrac{P \supseteq P' \quad [P']\ \mathsf{r}\ [Q'] \quad Q' \supseteq Q}{[P]\ \mathsf{r}\ [Q]}$	$\dfrac{P \subseteq P' \quad \{P'\}\ \mathsf{r}\ \{Q'\} \quad Q' \subseteq Q}{\{P\}\ \mathsf{r}\ \{Q\}}$
seq	$\dfrac{\langle\!\langle P \rangle\!\rangle\ \mathsf{r}_1\ \langle\!\langle R \rangle\!\rangle \quad \langle\!\langle R \rangle\!\rangle\ \mathsf{r}_2\ \langle\!\langle Q \rangle\!\rangle}{\langle\!\langle P \rangle\!\rangle\ \mathsf{r}_1; \mathsf{r}_2\ \langle\!\langle Q \rangle\!\rangle}$	$\dfrac{[P]\ \mathsf{r}_1\ [R] \quad [R]\ \mathsf{r}_2\ [Q]}{[P]\ \mathsf{r}_1; \mathsf{r}_2\ [Q]}$	$\dfrac{\{P\}\ \mathsf{r}_1\ \{R\} \quad \{R\}\ \mathsf{r}_2\ \{Q\}}{\{P\}\ \mathsf{r}_1; \mathsf{r}_2\ \{Q\}}$
choice	$\dfrac{\forall i \in \{1,2\} \quad \langle\!\langle P_i \rangle\!\rangle\ \mathsf{r}_i\ \langle\!\langle Q \rangle\!\rangle}{\langle\!\langle P_1 \cup P_2 \rangle\!\rangle\ \mathsf{r}_1 + \mathsf{r}_2\ \langle\!\langle Q \rangle\!\rangle}$	$\dfrac{\forall i \in \{1,2\} \quad [P]\ \mathsf{r}_i\ [Q_i]}{[P]\ \mathsf{r}_1 + \mathsf{r}_2\ [Q_1 \cup Q_2]}$	$\dfrac{\forall i \in \{1,2\} \quad \{P\}\ \mathsf{r}_i\ \{Q_i\}}{\{P\}\ \mathsf{r}_1 + \mathsf{r}_2\ \{Q_1 \cup Q_2\}}$
iter	$\dfrac{\forall n \geq 0 \,.\, \langle\!\langle Q_{n+1} \rangle\!\rangle\ \mathsf{r}\ \langle\!\langle Q_n \rangle\!\rangle}{\langle\!\langle \bigcup_{n \geq 0} Q_n \rangle\!\rangle\ \mathsf{r}^*\ \langle\!\langle Q_0 \rangle\!\rangle}$	$\dfrac{\forall n \geq 0 \,.\, [P_n]\ \mathsf{r}\ [P_{n+1}]}{[P_0]\ \mathsf{r}^*\ [\bigcup_{n \geq 0} P_n]}$	$\dfrac{\{P\}\ \mathsf{r}\ \{P\}}{\{P\}\ \mathsf{r}^*\ \{P\}}$
empty	$\overline{\langle\!\langle \emptyset \rangle\!\rangle\ \mathsf{r}\ \langle\!\langle Q \rangle\!\rangle}$	$\overline{[P]\ \mathsf{r}\ [\emptyset]}$	$\overline{\{\emptyset\}\ \mathsf{r}\ \{Q\}}$
disj	$\dfrac{\langle\!\langle P_1 \rangle\!\rangle\ \mathsf{r}\ \langle\!\langle Q_1 \rangle\!\rangle \quad \langle\!\langle P_2 \rangle\!\rangle\ \mathsf{r}\ \langle\!\langle Q_2 \rangle\!\rangle}{\langle\!\langle P_1 \cup P_2 \rangle\!\rangle\ \mathsf{r}\ \langle\!\langle Q_1 \cup Q_2 \rangle\!\rangle}$	$\dfrac{[P_1]\ \mathsf{r}\ [Q_1] \quad [P_2]\ \mathsf{r}\ [Q_2]}{[P_1 \cup P_2]\ \mathsf{r}\ [Q_1 \cup Q_2]}$	$\dfrac{\{P_1\}\ \mathsf{r}\ \{Q_1\} \quad \{P_2\}\ \mathsf{r}\ \{Q_2\}}{\{P_1 \cup P_2\}\ \mathsf{r}\ \{Q_1 \cup Q_2\}}$
iter0	$\overline{\langle\!\langle Q \rangle\!\rangle\ \mathsf{r}^*\ \langle\!\langle Q \rangle\!\rangle}$	$\overline{[P]\ \mathsf{r}^*\ [P]}$	unsound
unroll	$\dfrac{\langle\!\langle P \rangle\!\rangle\ \mathsf{r}^*; \mathsf{r}\ \langle\!\langle Q \rangle\!\rangle}{\langle\!\langle P \rangle\!\rangle\ \mathsf{r}^*\ \langle\!\langle Q \rangle\!\rangle}$	$\dfrac{[P]\ \mathsf{r}^*; \mathsf{r}\ [Q]}{[P]\ \mathsf{r}^*\ [Q]}$	unsound

Conclusions

Conclusions and future works

Many approaches to program (in)correctness.

- We sorted and compared some of these approaches.
- We designed SIL, a sound, complete and minimal proof system for backward under-approximation. To our knowledge, it is the first complete proof system designed for backward inference.

Conclusions and future works

Many approaches to program (in)correctness.

- We sorted and compared some of these approaches.
- We designed SIL, a sound, complete and minimal proof system for backward under-approximation. To our knowledge, it is the first complete proof system designed for backward inference.

Future works:

- Extend the schema (OL [Zilberstein et al. 2023], ESL [Maksimovic et al. 2023], UNTer [Raad et al. 2024], ...)
- Combine IL and SIL
- Backward LCL based on SIL?

Thanks for your attention!

Flavio Ascari

✉ flavio.ascari@phd.unipi.it


C. A. R. Hoare. *An Axiomatic Basis for Computer Programming*. Commun. ACM 12, 10 (1969)


P. W. O'Hearn. *Incorrectness logic*. POPL 2020


P. Cousot, R. Cousot and F. Logozzo. *Precondition Inference from Intermittent Assertions and Application to Contracts on Collections*. VMCAI 2011


E. de Vries and V. Koutavas. *Reverse Hoare Logic*. SEFM 2011


N. Zilberstein, D. Dreyer and A. Silva. *Outcome Logic: A Unifying Foundation for Correctness and Incorrectness Reasoning*. OOPSLA 2023


A. Raad, J. Berdine, H. Dang, D. Dreyer, P. W. O'Hearn and J. Villard. *Local Reasoning About the Presence of Bugs: Incorrectness Separation Logic*. CAV 2020


P. Maksimovic, C. Cronjäger, A. Lööw, J. Sutherland and P. Gardner. *Exact Separation Logic: Towards Bridging the Gap Between Verification and Bug-Finding*. ECOOP 2023


A. Raad, J. Vanegue and P. W. O'Hearn. *Compositional Non-Termination Proving*. Preprint


P. Cousot. *Calculational Design of [In]Correctness Transformational Program Logics by Abstract Interpretation*. POPL 2024