Principles of Abstract Interpretation

MIT press

Ch. **35**, Fixpoint checking

Patrick Cousot

pcousot.github.io

PrAbsInt@gmail.com   github.com/PrAbsInt/

These slides are available at
http://github.com/PrAbsInt/slides/slides-35--abstract-fixpoint-checking-PrAbsInt.pdf

Chapter 35

Ch. **35**, Fixpoint checking

- Many static analyzes are performed to check program properties which amount to proving that $\mathsf{lfp}^{\sqsubseteq}\, \mathscr{F}[\![\mathsf{P}]\!] \sqsubseteq P$ where property P is a specification.

- A common solution based on Park's fixpoint induction Theorem 24.1 consists in computing an inductive property I which is shown to be invariant $\mathscr{F}[\![\mathsf{P}]\!](I) \sqsubseteq I$ so that $\mathsf{lfp}^{\sqsubseteq}\, \mathscr{F}[\![\mathsf{P}]\!] \sqsubseteq I$ and stronger than P *i.e.* $I \sqsubseteq P$.

- A better solution is to use the specification P to improve the precision of the invariant I.

- The invariant I is inferred assuming the specification P does hold. Then the invariant I is checked to imply that $\mathsf{lfp}^{\sqsubseteq}\, \mathscr{F}[\![\mathsf{P}]\!] \sqsubseteq P$.

Concrete fixpoint checking

Concrete fixpoint checking

Theorem (35.1, Concrete fixpoint checking) Let $f \in \mathcal{L} \xrightarrow{\nearrow} \mathcal{L}$ be an increasing function on a complete lattice $\langle \mathcal{L}, \sqsubseteq, \bot, \top, \sqcap, \sqcup \rangle$ and $P \in \mathcal{L}$.

Then $\mathsf{lfp}^{\sqsubseteq} f \sqsubseteq P$ if and only if there exists $I \in \mathcal{L}$ such that $(f(I) \sqcap P) \sqsubseteq I$ and $f(I) \sqsubseteq P$.

- In static analysis, the advantage of Theorem 35.1 over Theorem 24.1 is that the invariant can be computed as an over-approximation of $\mathsf{lfp}^{\sqsubseteq} x \mapsto f(x) \sqcap P$ which is more precise that $\mathsf{lfp}^{\sqsubseteq} f$.
- For example the intersection with P may limit and reduce the extrapolation performed by a widening so improve the precision of the static analysis .

Proof of Theorem 35.1 I

Since f is increasing, $x \mapsto f(x) \sqcap P$ is also increasing so $\mathsf{lfp}^{\sqsubseteq} f$ and $\mathsf{lfp}^{\sqsubseteq} x \mapsto f(x) \sqcap P$ do exist by Tarski's fixpoint Theorem 15.6.

($\Leftarrow$, soundness)

$$
\begin{array}{ll}
& (f(I) \sqcap P) \sqsubseteq I \wedge f(I) \sqsubseteq P \\
\Rightarrow & (f(I) \sqcap P) \sqsubseteq I \wedge f(I) \sqcap P = f(I) \wedge f(I) \sqsubseteq P \quad \wr \text{def. } \sqsubseteq \wr \\
\Rightarrow & f(I) \sqsubseteq I \wedge f(I) \sqsubseteq P \quad \wr \text{substitution} \wr \\
\Rightarrow & \mathsf{lfp}^{\sqsubseteq} f \sqsubseteq I \wedge f(I) \sqsubseteq P \quad \wr \text{Tarski's fixpoint Theorem 15.6} \wr \\
\Rightarrow & \mathsf{lfp}^{\sqsubseteq} f \sqsubseteq P \quad \wr \text{since } f \text{ is increasing so } \mathsf{lfp}^{\sqsubseteq} f = f(\mathsf{lfp}^{\sqsubseteq} f) \sqsubseteq f(I) \sqsubseteq P \text{ and transitivity} \wr
\end{array}
$$

(⇒, completeness)

$$
\begin{array}{ll}
 & \mathsf{lfp}^{\sqsubseteq} f \sqsubseteq P \\
\Rightarrow & \mathsf{lfp}^{\sqsubseteq} f \sqcap P = \mathsf{lfp}^{\sqsubseteq} f \wedge \mathsf{lfp}^{\sqsubseteq} f \sqsubseteq P \qquad \text{(def. } \sqsubseteq \text{ and } \sqcap\text{)} \\
\Rightarrow & f(\mathsf{lfp}^{\sqsubseteq} f) \sqcap P \sqsubseteq \mathsf{lfp}^{\sqsubseteq} f \wedge \mathsf{lfp}^{\sqsubseteq} f \sqsubseteq P \qquad \text{(fixpoint property and } \sqsubseteq \text{ reflexive)} \\
\Rightarrow & f(I) \sqcap P \sqsubseteq I \wedge I \sqsubseteq P \qquad \text{(take } I = \mathsf{lfp}^{\sqsubseteq} f\text{)} \quad \square
\end{array}
$$

Abstract fixpoint checking

Abstract fixpoint checking

Theorem (35.5. Abstract fixpoint checking) Let $f \in L \xrightarrow{uc} L$ be an upper continuous function on a cpo $\langle L, \sqsubseteq, \bot\rangle$ with infimum $\bot$.

Let $\langle \overline{L}, \overline{\sqsubseteq}\rangle$ be an abstract domain with increasing concretization $\gamma \in \overline{L} \xrightarrow{\nearrow} L$.

Let $\overline{f} \in \overline{L} \rightarrow \overline{L}$ be such that $f \circ \gamma \mathrel{\dot{\sqsubseteq}} \gamma \circ \overline{f}$ (semi-commutation).

Let $\overline{P} \in \overline{L}$ be an abstract specification.

If, assuming the glb $\overline{\sqcap}$ exists, $\exists \overline{I} \in \overline{L} \,.\, \overline{f}(\overline{I}) \mathrel{\overline{\sqcap}} \overline{P} \mathrel{\overline{\sqsubseteq}} \overline{I} \wedge \overline{f}(\overline{I}) \mathrel{\overline{\sqsubseteq}} \overline{P}$ then $\mathsf{lfp}^{\sqsubseteq} f \sqsubseteq \gamma(\overline{P})$.

Proof of Theorem 35.5 I

$$
\begin{array}{lll}
& \overline{f}(\overline{I}) \sqcap \overline{P} \mathrel{\overline{\sqsubseteq}} \overline{I} \wedge \overline{f}(\overline{I}) \mathrel{\overline{\sqsubseteq}} \overline{P} & \\
\Rightarrow & \overline{f}(\overline{I}) \mathrel{\overline{\sqcap}} \overline{P} \mathrel{\overline{\sqsubseteq}} \overline{I} \wedge \overline{f}(\overline{I}) \mathrel{\overline{\sqcap}} \overline{P} = \overline{f}(\overline{I}) \wedge \overline{f}(\overline{I}) \mathrel{\overline{\sqsubseteq}} \overline{P} & \text{(def. } \overline{\sqsubseteq}\text{)} \\
\Rightarrow & \overline{f}(\overline{I}) \mathrel{\overline{\sqsubseteq}} \overline{I} \wedge \overline{f}(\overline{I}) \mathrel{\overline{\sqsubseteq}} \overline{P} & \text{(since } \overline{f}(\overline{I}) \mathrel{\overline{\sqcap}} \overline{P} = \overline{f}(\overline{I})\text{)} \\
\Rightarrow & \gamma(\overline{f}(\overline{I})) \sqsubseteq \gamma(\overline{I}) \wedge \gamma(\overline{f}(\overline{I})) \sqsubseteq \gamma(\overline{P}) & \text{(}\gamma\text{ increasing)} \\
\Rightarrow & f(\gamma(\overline{I})) \sqsubseteq \gamma(\overline{I}) \wedge f(\gamma(\overline{I})) \sqsubseteq \gamma(\overline{P}) & \text{(semi-commutation and transitivity)} \\
\Rightarrow & \mathsf{lfp}^{\sqsubseteq} f \sqsubseteq \gamma(\overline{P}) & \text{(Theorem 35.1 with } I = \gamma(\overline{I}) \text{ and } P = \gamma(\overline{P})\text{)} \quad \square
\end{array}
$$

Invariants may not help enough

- Theorem 35.1 shows that adding information about the program behavior can only help the analysis.
- So if a static analysis is not precise enough *e.g.* because of excessive extrapolations or interpolations, adding a specification of what the static analyzer should infer is sound and can only help.
- This is however in general insufficient.
- The main reason is that if the specification P is given in the concrete then it will be abstracted in $\overline{P}$ in the abstract domain where the concrete information may be lost.
- Moreover, if the specification $\overline{P}$ is given in the abstract then it might not be inductive.

Conclusion

- Astrée [Bertrane, P. Cousot, R. Cousot, Feret, Mauborgne, Miné, and Rival, 2015] is based on Theorem 35.5: the static analysis is done assuming the specification holds and then it is verified that the specification does hold.
- [P. Cousot, 2000] considers equivalent forms of Theorems 35.5 and 35.5 based on duality.

`www.absint.com/astree/index.htm`

Bibliography I

Bertrane, Julien, Patrick Cousot, Radhia Cousot, Jérôme Feret, Laurent Mauborgne, Antoine Miné, and Xavier Rival (2015). "Static Analysis and Verification of Aerospace Software by Abstract Interpretation". *Foundations and Trends in Programming Languages* 2.2-3, pp. 71–190.

Cousot, Patrick (2000). "Partial Completeness of Abstract Fixpoint Checking". In: *SARA*. Vol. 1864. Lecture Notes in Computer Science. Springer, pp. 1–25.

Home work

Read Ch. **35** "Fixpoint checking" of

Principles of Abstract Interpretation
Patrick Cousot
MIT Press

The End, Thank you