Principles of Abstract Interpretation

MIT press

Ch. **29**, Reduction

Patrick Cousot

pcousot.github.io

PrAbsInt@gmail.com   github.com/PrAbsInt/

These slides are available at
http://github.com/PrAbsInt/slides/slides-29--reduction-PrAbsInt.pdf

Chapter 29

Ch. **29**, Reduction

Objective

- study the reduction idea
- apply it to the analysis (28.38) of boolean expressions.

Reduction

Exercise 15.11

Given an increasing function $f \in L \xrightarrow{\nearrow} L$ on a complete lattice $\langle L, \sqsubseteq, \bot, \top, \sqcap, \sqcup \rangle$ and a prefixpoint $a \in L$ such that $a \sqsubseteq f(a)$, show that $\mathsf{lfp}_a^{\sqsubseteq} f = \bigsqcap\{x \in L \mid a \sqsubseteq x \wedge f(x) \sqsubseteq x\}$ is the least fixpoint of f greater than or equal to a. $\square$

Proof

- Define $L_a \triangleq \{x \in L \mid a \sqsubseteq x\}$
- $\langle L_a, \sqsubseteq, a, \top, \sqcap, \sqcup \rangle$ is a complete lattice
- $f \in L_a \xrightarrow{\nearrow} L_a$ since $x \in L_a$ implies $a \sqsubseteq x$ so $a \sqsubseteq f(a) \sqsubseteq f(x)$ proving $f(x) \in L_a$
- Applying Tarski's fixpoint Theorem 15.6 to f on L_a

$$\mathsf{lfp}_a^{\sqsubseteq} f$$

$$= \bigsqcap\{x \in L_a \mid f(x) \sqsubseteq x\}$$

$$= \bigsqcap\{x \in L \mid a \sqsubseteq x \wedge f(x) \sqsubseteq x\}$$

$\square$

Exercise 10.5

In a complete lattice $\langle \mathbb{P}, \sqsubseteq, \bot, \top, \sqcup \rangle$, if $X, Y \in \wp(\mathbb{P})$ and $X \subseteq Y$ then $\sqcup X \sqsubseteq \sqcup Y$.

Proof By def. $\subseteq$, $\forall x \in X \,.\, x \in Y$ so $x \sqsubseteq \sqcup Y$ by def. lub $\sqcup$.

It follows that $\sqcup Y$ is an upper bound of X so $\sqcup X \sqsubseteq \sqcup Y$ by def. lub $\sqcup$. $\square$

In a complete lattice $\langle \mathbb{P}, \sqsubseteq, \bot, \top, \sqcup \rangle$, if $X, Y \in \wp(\mathbb{P})$ and $X \subseteq Y$ then $\sqcap Y \sqsubseteq \sqcap X$.

Proof By duality. $\square$

Pointwise fixpoint over-approximation

Theorem (18.7, pointwise fixpoint over-approximation) Assume that $\langle C, \sqsubseteq, \bot, \top, \sqcup, \sqcap\rangle$ is a complete lattice, $f, g \in C \xrightarrow{\nearrow} C$ are increasing, and $f \mathrel{\dot{\sqsubseteq}} g$ then $\mathsf{lfp}^{\sqsubseteq} f \sqsubseteq \mathsf{lfp}^{\sqsubseteq} g$.

Proof

- By $f \mathrel{\dot{\sqsubseteq}} g$, for all $x \in C$, $g(x) \sqsubseteq x$ implies $f(x) \sqsubseteq x$ so $\{x \in C \mid g(x) \sqsubseteq x\} \subseteq \{x \in C \mid f(x) \sqsubseteq x\}$
- so, by Tarski's fixpoint Theorem 15.6 and dual of Exercise 10.5, $\mathsf{lfp}^{\sqsubseteq} f = \bigsqcap\{x \in C \mid f(x) \sqsubseteq x\} \sqsubseteq \bigsqcap\{x \in C \mid g(x) \sqsubseteq x\} = \mathsf{lfp}^{\sqsubseteq} g$. $\square$

Lemma (29.1, [P. Cousot and R. Cousot, 1979a])

- Let $g \in L \to L$ by an increasing [1] and extensive [2] operator on a complete lattice $\langle L, \sqsubseteq, \bot, \top, \sqcap, \sqcup\rangle$.
- Define $\hat{\boldsymbol{\rho}}(x) \triangleq \mathsf{lfp}^{\sqsubseteq}_{x}\, g$ where $\mathsf{lfp}^{\sqsubseteq}_{a}\, f$ is the least fixpoint of f greater than or equal to a, if any.
- Then $\hat{\boldsymbol{\rho}}$ is the $\dot{\sqsubseteq}$-smallest upper closure operator[3] pointwise greater than or equal to g [4,5].

[1] $\forall x, y \in L \,.\, (x \sqsubseteq y) \Rightarrow g(x) \sqsubseteq g(y)$.

[2] $\forall x \in L \,.\, x \sqsubseteq g(x)$.

[3] increasing, extensive, and idempotent ($\hat{\boldsymbol{\rho}} \circ \hat{\boldsymbol{\rho}} = \hat{\boldsymbol{\rho}}$)

[4] $\forall x \in L \,.\, g(x) \sqsubseteq \hat{\boldsymbol{\rho}}(x)$.

[5] *i.e.* $g \,\dot{\sqsubseteq}\, \hat{\boldsymbol{\rho}}$ and if $\hat{\boldsymbol{\rho}}'$ is also increasing, extensive, and idempotent such that $g \,\dot{\sqsubseteq}\, \hat{\boldsymbol{\rho}}'$ then $g \,\dot{\sqsubseteq}\, \hat{\boldsymbol{\rho}} \,\dot{\sqsubseteq}\, \hat{\boldsymbol{\rho}}'$

Proof of Lemma 29.1 — (1) — Let us first prove that $\hat{\boldsymbol{\rho}}$ is an upper closure operator.

- By the variant Exercise 15.11 of Tarski's fixpoint Theorem 15.6 and g reductive so $g(x) \sqsubseteq x$, we have $\hat{\boldsymbol{\rho}}(x) = \mathsf{lfp}_x^{\sqsubseteq} g = \bigsqcap\{y \mid g(y) \sqsubseteq y \wedge x \sqsubseteq y\}$.
- $\hat{\boldsymbol{\rho}}$ is obviously extensive since all y in $\{y \mid g(y) \sqsubseteq y \wedge x \sqsubseteq y\}$ are an upper bound of x hence $x \sqsubseteq \bigsqcap\{y \mid g(y) \sqsubseteq y \wedge x \sqsubseteq y\} = \mathsf{lfp}_x^{\sqsubseteq} g = \hat{\boldsymbol{\rho}}(x)$ by def. of the lub $\bigsqcap$.
- If $x \sqsubseteq x'$ then $\{y \mid g(y) \sqsubseteq y \wedge x' \sqsubseteq y\} \subseteq \{y \mid g(y) \sqsubseteq y \wedge x \sqsubseteq y\}$ so, by the order-dual of Exercise 10.5, $\hat{\boldsymbol{\rho}}(x) = \mathsf{lfp}_x^{\sqsubseteq} g = \bigsqcap\{y \mid g(y) \sqsubseteq y \wedge x \sqsubseteq y\} \sqsubseteq \bigsqcap\{y \mid g(y) \sqsubseteq y \wedge x' \sqsubseteq y\} = \mathsf{lfp}_{x'}^{\sqsubseteq} g = \hat{\boldsymbol{\rho}}(x')$, proving that $\hat{\boldsymbol{\rho}}$ is increasing.
- By extension $x \sqsubseteq \hat{\boldsymbol{\rho}}(x)$ so $\hat{\boldsymbol{\rho}}(x) \sqsubseteq \hat{\boldsymbol{\rho}}(\hat{\boldsymbol{\rho}}(x))$ by increasingness.
- Moreover, $\hat{\boldsymbol{\rho}}(x) = \mathsf{lfp}_x^{\sqsubseteq} g$ is a fixpoint of g so $g(\hat{\boldsymbol{\rho}}(x)) = \hat{\boldsymbol{\rho}}(x)$ $g(\hat{\boldsymbol{\rho}}(x)) \sqsubseteq \hat{\boldsymbol{\rho}}(x)$ by reflexivity. It follows that $\hat{\boldsymbol{\rho}}(x) \in \{y \mid g(y) \sqsubseteq y \wedge \hat{\boldsymbol{\rho}}(x) \sqsubseteq y\}$ proving that $\hat{\boldsymbol{\rho}}(\hat{\boldsymbol{\rho}}(x)) = \mathsf{lfp}_{\hat{\boldsymbol{\rho}}(x)}^{\sqsubseteq} g = \bigsqcap\{y \mid g(y) \sqsubseteq y \wedge \hat{\boldsymbol{\rho}}(x) \sqsubseteq y\} \sqsubseteq \hat{\boldsymbol{\rho}}(x)$.
- By antisymmetry, $\hat{\boldsymbol{\rho}}(\hat{\boldsymbol{\rho}}(x)) = \hat{\boldsymbol{\rho}}(x)$ proving idempotency. $\square$

— (2) — Let us prove that $g \mathrel{\dot{\sqsubseteq}} \hat{\boldsymbol{\rho}}$ that is $\forall x \in L \mathrel{.} g(x) \sqsubseteq \hat{\boldsymbol{\rho}}(x)$ pointwise,

- observe that, by def. of the least fixpoint, $x \sqsubseteq \mathsf{lfp}_x^{\sqsubseteq}\, g$
- So, by increasingness, $g(x) \sqsubseteq g(\mathsf{lfp}_x^{\sqsubseteq}\, g) = \mathsf{lfp}_x^{\sqsubseteq}\, g \triangleq \hat{\boldsymbol{\rho}}(x)$.

— (3) — Let ρ' be an upper closure operator greater that of equal to g *i.e.* $g \mathrel{\dot{\sqsubseteq}} \rho'$.

- We have $x \sqsubseteq \rho'(x) = \rho'(\rho'(x))$
- so $x \sqsubseteq \mathsf{lfp}_x^{\sqsubseteq}\, \rho' \sqsubseteq \rho'(x)$
- hence $\rho'(x) \sqsubseteq \rho'(\mathsf{lfp}_x^{\sqsubseteq}\, \rho') = \mathsf{lfp}_x^{\sqsubseteq}\, \rho' \sqsubseteq \rho'(\rho'(x)) = \rho'(x)$ by increasingness and fixpoint property,
- proving $\mathsf{lfp}_x^{\sqsubseteq}\, \rho' = \rho'(x)$ by antisymmetry.
- By increasingness of g and ρ' and Theorem 18.7, $\hat{\boldsymbol{\rho}}(x) = \mathsf{lfp}_x^{\sqsubseteq}\, g \sqsubseteq \mathsf{lfp}_x^{\sqsubseteq}\, \rho' = \rho'(x)$,
- proving $\hat{\boldsymbol{\rho}} \mathrel{\dot{\sqsubseteq}} \rho'$ pointwise
- and so that $\hat{\boldsymbol{\rho}}$ is the smallest upper closure operator pointwise greater than or equal to g. $\square$

Theorem (29.2) Let $\langle C, \preceq, 0, 1, \curlywedge, \curlyvee\rangle$ and $\langle A, \sqsubseteq, \bot, \top, \sqcap, \sqcup\rangle$ be complete lattices such that $\langle C, \preceq\rangle \xleftrightarrow[\alpha]{\gamma} \langle A, \sqsubseteq\rangle$, $f \in C \to C$ be a lower closure operator on C, and $g \in A \to A$ be an increasing and reductive operator on A such that $\alpha \circ f \circ \gamma \mathrel{\dot{\sqsubseteq}} g$. Then the reduction $\check{\boldsymbol{\rho}}(x) \triangleq \mathsf{gfp}_x^{\sqsubseteq}\, g$ satisfies $\alpha \circ f \circ \gamma \mathrel{\dot{\sqsubseteq}} \check{\boldsymbol{\rho}} \mathrel{\dot{\sqsubseteq}} g$.

t

Proof of Theorem 29.2—

- We have $\alpha \circ f \circ \gamma(x) \sqsubseteq \alpha \circ f \circ f \circ \gamma(x)$ since f is idempotent.
- Moreover $\alpha \circ f \circ f \circ \gamma(x) \sqsubseteq \alpha \circ f \circ \gamma \circ \alpha \circ f \circ \gamma(x)$ since $\gamma \circ \alpha$ is extensive, α and f are increasing.
- By transitivity, $\alpha \circ f \circ \gamma(x) \sqsubseteq \alpha \circ f \circ \gamma \circ \alpha \circ f \circ \gamma(x)$.
- We have $f \circ \gamma(x) \preceq \gamma(x)$ since f is reductive
- so $\alpha \circ f \circ \gamma(x) \sqsubseteq x$ by def. of the Galois connection.
- It follows that $\alpha \circ f \circ \gamma(x) \in \{y \mid y \sqsubseteq \alpha \circ f \circ \gamma(y) \wedge y \sqsubseteq x\}$
- proving $\alpha \circ f \circ \gamma(x) \sqsubseteq \bigsqcup\{y \mid y \sqsubseteq \alpha \circ f \circ \gamma(y) \wedge y \sqsubseteq x\} = \mathsf{gfp}_x^{\sqsubseteq}\, \alpha \circ f \circ \gamma$ by def. lub $\sqcup$ and the order-dual ofExercise 15.11
- By self-duality of $\dot{\sqsubseteq}$ and the dual of Theorem 18.7, we have $\alpha \circ f \circ \gamma \mathrel{\dot{\sqsubseteq}} g$ implies $\mathsf{gfp}_x^{\sqsubseteq}\, \alpha \circ f \circ \gamma \sqsubseteq \mathsf{gfp}_x^{\sqsubseteq}\, g = \check{\boldsymbol{\rho}}(x)$.

- By transitivity, we conclude that $\alpha \circ f \circ \gamma(x) \sqsubseteq \check{\boldsymbol{\rho}}(x)$ *i.e.* $\alpha \circ f \circ \gamma \mathrel{\dot{\sqsubseteq}} \check{\boldsymbol{\rho}}$ pointwise.

— By the dual of Lemma 29.1, $\check{\boldsymbol{\rho}}$ is the largest lower closure operator pointwise less than or equal to g. So $\check{\boldsymbol{\rho}} \mathrel{\dot{\sqsubseteq}} g$ and therefore $\alpha \circ f \circ \gamma \mathrel{\dot{\sqsubseteq}} \check{\boldsymbol{\rho}} \mathrel{\dot{\sqsubseteq}} g$. $\square$

Corollary (29.3) Let g^n be the iterates of g *i.e.* $g^1 \triangleq g$ and $g^{n+1} \triangleq g \circ g^n$. Then under the hypotheses of Theorem 29.2, $\alpha \circ f \circ \gamma \mathrel{\dot{\sqsubseteq}} \check{\boldsymbol{\rho}} \mathrel{\dot{\sqsubseteq}} g^n \mathrel{\dot{\sqsubseteq}} g$.

Proof of Corollary 29.3 By recurrence on $n \in \mathbb{N}^+$.

- Follows from Theorem 29.2 and reflexivity for $n = 1$.
- For the inductive step, g is increasing so $g \circ \check{\boldsymbol{\rho}} \mathrel{\dot{\sqsubseteq}} g \circ g^n = g^{n+1} \mathrel{\dot{\sqsubseteq}} g \circ g \mathrel{\dot{\sqsubseteq}} g$ since g is reductive and increasing.
- Moreover the dual proof of Lemma 29.1 shows that $\check{\boldsymbol{\rho}}$ is a fixpoint of g so $g \circ \check{\boldsymbol{\rho}} = \check{\boldsymbol{\rho}}$, proving $\alpha \circ f \circ \gamma \mathrel{\dot{\sqsubseteq}} \check{\boldsymbol{\rho}} \mathrel{\dot{\sqsubseteq}} g^{n+1} \mathrel{\dot{\sqsubseteq}} g$ by reflexivity and transitivity. $\square$

Section **29.2**, Test reduction

Test reduction

- By Exercise 28.37, $\mathsf{test}^{\times}[\![\mathsf{B}]\!]$ is a lower closure operator and, by Exercise 28.41, $\mathsf{test}^{\between}[\![\mathsf{B}]\!]$ is increasing and reductive so we can apply Corollary 29.3 and iterate $\mathsf{test}^{\between}[\![\mathsf{B}]\!]$.
- If the abstract domain has no infinite strictly decreasing chains, this iteration will stop at the greatest fixpoint.
- Otherwise, convergence can be enforced by a narrowing (introduced in Definition 34.14 *e.g.* by stopping after any number of iterations).

Example: without local iterations

For example, the parity analysis of the program `y=1;if(z<0) z=x; else z=y; if ((x==z) nand (y==z)) x=1; else y=2;` with (28.38) is

```
 l1: [x:e; y:e; z:e] y = 1;
if l2: (z < 0) [x:e; y:o; z:e]
   l3: [x:e; y:o; z:e] z = x;
 else
   l4: [x:e; y:o; z:e] z = y;
if l5: ((x == z) nand (y == z)) [x:e; y:o; z:T]
   l6: [x:e; y:o; z:T] x = 1;
 else
   l7: [x:e; y:o; z:_|_] y = 2;
l8: [x:T; y:T; z:T]
```

i.e. `z` cannot be both even and odd when `((x == z) nand (y == z))` is false that is $(x = z) \wedge (y = z)$.

Example: with local iterations

With local iterations for tests (28.38), this information is propagated to x and y

```
 l1: [x:e; y:e; z:e] y = 1;
if l2: (z < 0) [x:e; y:o; z:e]
   l3: [x:e; y:o; z:e] z = x;
 else
   l4: [x:e; y:o; z:e] z = y;
if l5: ((x == z) nand (y == z)) [x:e; y:o; z:T]
   l6: [x:e; y:o; z:T] x = 1;
 else
   l7: [x:_|_; y:_|_; z:_|_] y = 2;
l8: [x:o; y:o; z:T]
```

See Section **33.7** for further examples.

Conclusion

Reduction for cartesian abstractions

- Reductions can improve the precision of analyzes. This was shown for local iterations for tests on cartesian abstract domains.
- This is also useful for type inference.
- For example Ada [Ichbiah, Krieg-Brückner, Wichmann, Barnes, Roubine, and Héliard, 1979] allows for user-defined overloading of predefined arithmetic operators not only on the basis of argument types (as previously in Algol68 [Wijngaarden, Mailloux, Peck, Koster, Sintzoff, Lindsey, Meertens, and Fisker, 1975]), but also result types.
- So Bernd Krieg-Brückner designed an iterated reduction algorithm of the top-down type inference of the result type from the argument types and a bottom-up type inference of the argument types from the type inference that was adopted in Ada [Ichbiah, Krieg-Brückner, Wichmann, Barnes, Roubine, and Héliard, 1979, Section 7.5.1].
- This extends to higher-order types as in the CASL formal specification language [Astesiano, Bidoit, Kirchner, Krieg-Brückner, Mosses, Sannella, and Tarlecki, 2002].

Reduction for relational abstractions

- However, the precision gain for relational domains may not be that spectacular when boolean operators can be taken into account precisely.
- This is why *e.g*. the Astrée static analyzer [Bertrane, P. Cousot, R. Cousot, Feret, Mauborgne, Miné, and Rival, 2015] does not use local iterations for tests.
- The idea of iterating an increasing and reductive abstract operator to get a more precise lower closure abstract operator was used
  - to define the "reduced product" of [P. Cousot and R. Cousot, 1979b]
  - in the "local decreasing iterations" [Granger, 1992] to handle backward assignments and conditionals
  - (the same idea was later exploited in logic program analysis under the name of "reexecution" [Le Charlier and Van Hentenryck, 1992]).

Bibliography I

Astesiano, Egidio, Michel Bidoit, Hélène Kirchner, Bernd Krieg-Brückner, Peter D. Mosses, Donald Sannella, and Andrzej Tarlecki (2002). “CASL: the Common Algebraic Specification Language”. *Theor. Comput. Sci.* 286.2, pp. 153–196.

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 and Radhia Cousot (1979a). “A constructive characterization of the lattices of all retractions, pre-closure, quasi-closure and closure operators on a complete lattice”. *Portugaliæ Mathematica* 38.2, pp. 185–198.

— (1979b). “Systematic Design of Program Analysis Frameworks”. In: *POPL*. ACM Press, pp. 269–282.

Bibliography II

Granger, Philippe (1992). "Improving the Results of Static Analyses Programs by Local Decreasing Iteration". In: *FSTTCS*. Vol. 652. Lecture Notes in Computer Science. Springer, pp. 68–79.

Ichbiah, Jean D., Bernd Krieg-Brückner, Brian A. Wichmann, John G. P. Barnes, Olivier Roubine, and Jean-Claude Héliard (1979). "Rationale for the design of the Ada programming language". *SIGPLAN Notices* 14.6b, pp. 1–261.

Le Charlier, Baudouin and Pascal Van Hentenryck (1992). "Reexecution in Abstract Interpretation of Prolog". In: *JICSLP*. MIT Press, pp. 750–764.

Wijngaarden, Adriaan van, B. J. Mailloux, J. E. L. Peck, Cornelis H. A. Koster, Michel Sintzoff, C. H. Lindsey, Lambert G. L. T. Meertens, and R. G. Fisker (1975). "Revised Report on the Algorithmic Language ALGOL 68". *Acta Inf.* 5, pp. 1–236.

Home work

Read Ch. **29** "Reduction" of

Principles of Abstract Interpretation
Patrick Cousot
MIT Press

The End, Thank you