Principles of Abstract Interpretation

MIT press

Ch. **21**, Abstract domain and abstract structural semantics

Patrick Cousot

pcousot.github.io

PrAbsInt@gmail.com github.com/PrAbsInt/

These slides are available at
http://github.com/PrAbsInt/slides/slides/slides-21--abstract-domain-best-abstraction-abstract-interpreter-PrAbsInt.pdf

Design of a verification/analysis method for a programming language by abstract interpretation

- Define the syntax and operational semantics of the language
- Define program properties and the collecting semantics
- Define an abstraction of properties (preferably by a Galois connection)
- Calculate a sound (and possibly complete) abstract semantics by abstraction of the collecting semantics ← this chapter

generic abstract properties and semantics

- Define an abstract inductive proof method/analysis algorithm

Chapter 21

Ch. **21**, Abstract domain and abstract structural semantics

Semantics

Pointwise prefix trace semantics of Section **6.12**

- Given a set $\mathcal{R}_0$ of initial traces, the pointwise prefix trace semantics $\dot{\mathcal{S}}^*[\![\mathtt{S}]\!]\,\mathcal{R}_0\,\ell$ is the set of the prefix traces starting from $\mathcal{R}_0$ and arriving at program label ℓ

$$\begin{array}{rcl} \dot{\mathcal{S}}^*[\![\mathtt{S}]\!] & \in & \wp(\mathbb{T}^+) \dot{\longrightarrow} (\mathbb{L} \to \wp(\mathbb{T}^+)) \\ \dot{\mathcal{S}}^*[\![\mathtt{S}]\!]\,\mathcal{R}_0\,\ell & \triangleq & \{\pi_0\ell_0\pi_1\ell_1 \mid \pi_0\ell_0 \in \mathcal{R}_0 \wedge \ell_0\pi_1\ell_1 \in \dot{\mathcal{S}}^*[\![\mathtt{S}]\!](\pi_0\ell_0) \wedge \ell_1 = \ell\} \end{array} \tag{6.47}$$

Relational reachability semantics of Section **19.1.2**

- Given a relation $\mathcal{R}_0$ between initial environments, the relational reachability semantics $\vec{\mathcal{S}^{\mathsf{R}}}[\![\mathtt{s}]\!]\,\mathcal{R}_0\,\ell$ is the relation between initial environments and those when arriving at program label ℓ

$$\begin{array}{rcll}
\vec{\mathcal{S}^{\mathsf{R}}}[\![\mathtt{s}]\!] & \in & \wp(\mathbb{E}\mathtt{v}\times\mathbb{E}\mathtt{v})\to(\mathbb{L}\to\wp(\mathbb{E}\mathtt{v}\times\mathbb{E}\mathtt{v})) & \text{(Section } \mathbf{19.1.3}\text{)}\\
\vec{\mathcal{S}^{\mathsf{R}}}[\![\mathtt{s}]\!]\,\mathcal{R}_0\,\ell & \triangleq & \{\langle\rho_0,\,\boldsymbol{\varrho}(\pi_0\ell_0\pi_1\ell')\rangle \mid \langle\rho_0,\,\boldsymbol{\varrho}(\pi_0\ell_0)\rangle\in\mathcal{R}_0\wedge\\
& & \qquad\qquad \exists\pi_2\,.\,\ell_0\pi_1\ell'\pi_2\in\mathcal{S}^*[\![\mathtt{s}]\!](\pi_0\ell_0)\wedge\ell'=\ell\}\\
& = & \ddot{\alpha}_{\boldsymbol{\varrho}}(\dot{\mathcal{S}^*}[\![\mathtt{s}]\!])\,\mathcal{R}_0\,\ell
\end{array}$$

Assertional reachability semantics of Section **19.1.1**

- Given a set $\mathcal{R}_0$ of initial environments, the assertional reachability semantics $\boldsymbol{\mathcal{S}}^{\vec{r}}[\![\mathtt{S}]\!]\,\mathcal{R}_0\,\ell$ is the set of environments reachable from $\mathcal{R}_0$ when arriving at program label ℓ

$$\begin{array}{rcll} \boldsymbol{\mathcal{S}}^{\vec{r}}[\![\mathtt{S}]\!] & \in & \wp(\mathbb{E}\mathtt{v}) \to (\mathbb{L} \to \wp(\mathbb{E}\mathtt{v})) & \text{(Section } \mathbf{19.1.3}\text{)} \\ \boldsymbol{\mathcal{S}}^{\vec{r}}[\![\mathtt{S}]\!]\,\mathcal{R}_0\,\ell & \triangleq & \{\boldsymbol{\varrho}(\pi_0\ell_0\pi_1\ell') \mid \boldsymbol{\varrho}(\pi_0\ell_0) \in \mathcal{R}_0 \wedge \exists \pi_2 \,.\, \ell_0\pi_1\ell'\pi_2 \in \boldsymbol{\mathcal{S}}^*[\![\mathtt{S}]\!](\pi_0\ell_0) \wedge \ell' = \ell\} & \\ & = & \ddot{\alpha}_2(\boldsymbol{\mathcal{S}}^{\vec{\mathtt{R}}}[\![\mathtt{S}]\!])\,\mathcal{R}_0\,\ell & \end{array}$$

Abstract domain

Common structure

The domain of properties, inclusion (*i.e.* logical implication), and the structural definitions of the semantics have the following common structure.

semantics	prefix trace $\widehat{\mathcal{S}}^{*}$	reachability $\widehat{\mathcal{S}}^{\vec{r}}$	abstract $\widehat{\mathcal{S}}^{\natural}$
	$\wp(\mathbb{T}^+) \rightarrow\!\!\!\!\!\rightarrow (\mathbb{L} \longrightarrow \wp(\mathbb{T}^+))$	$\wp(\mathbb{E}\mathbb{v}) \rightarrow\!\!\!\!\!\rightarrow (\mathbb{L} \longrightarrow \wp(\mathbb{E}\mathbb{v}))$	$\mathbb{P}^{\natural} \rightarrow\!\!\!\!\!\rightarrow (\mathbb{L} \longrightarrow \mathbb{P}^{\natural})$
	Chapters **6** and **17**	Chapter **19**	Chapter **21**
domain	$\wp(\mathbb{T}^+)$	$\wp(\mathbb{E}\mathbb{v})$	$\mathbb{P}^{\natural}$
inclusion	$\subseteq$	$\subseteq$	$\sqsubseteq^{\natural}$
abstraction	$\mathbb{1}_{\wp(\mathbb{T}^+)}$ [1]	$\ddot{\alpha}_{\boldsymbol{\varrho}}$	$\alpha_{\natural}$
infimum	$\varnothing$	$\varnothing$	$\bot^{\natural}$
join	$\cup$	$\cup$	$\sqcup^{\natural}$
assignment	$\mathrm{assign}^{*}[\![\mathrm{x}, \mathrm{A}]\!]$	$\mathrm{assign}^{\vec{r}}[\![\mathrm{x}, \mathrm{A}]\!]$	$\mathrm{assign}^{\natural}[\![\mathrm{x}, \mathrm{A}]\!]$
test	$\mathrm{test}^{*}[\![\mathrm{B}]\!]$	$\mathrm{test}^{\vec{r}}[\![\mathrm{B}]\!]$	$\mathrm{test}^{\natural}[\![\mathrm{B}]\!]$
	$\overline{\mathrm{test}}^{*}[\![\mathrm{B}]\!]$	$\overline{\mathrm{test}}^{\vec{r}}[\![\mathrm{B}]\!]$	$\overline{\mathrm{test}}^{\natural}[\![\mathrm{B}]\!]$

[1] $\mathbb{1}_S \triangleq x \in S \mapsto x$ is the identity function on the set S.

Definition (21.1, Domain well-definedness) We say that a domain

$$\mathbb{D}^\natural \quad \triangleq \quad \langle \mathbb{P}^\natural, \sqsubseteq^\natural, \bot^\natural, \sqcup^\natural, \mathtt{assign}_\natural[\![\mathtt{x}, \mathtt{A}]\!], \mathtt{test}^\natural[\![\mathtt{B}]\!], \overline{\mathtt{test}}^\natural[\![\mathtt{B}]\!] \rangle$$

is *well-defined* when $\langle \mathbb{P}^\natural, \sqsubseteq^\natural \rangle$ is a poset of properties with infimum $\bot^\natural$, the lub $\sqcup^\natural$ is well-defined both for pairs of properties and for $\sqsubseteq^\natural$-increasing chains (so $\langle \mathbb{P}^\natural, \sqsubseteq^\natural \rangle$ is a join-lattice and a cpo), the assignment $\mathtt{assign}^\natural$ is well-defined in $(\mathcal{V} \times \mathbb{E}) \to \mathbb{P}^\natural \dashrightarrow \mathbb{P}^\natural$, and the tests $\mathtt{test}^\natural[\![\mathtt{B}]\!]$ and $\overline{\mathtt{test}}^\natural[\![\mathtt{B}]\!]$ are well-defined in $\mathbb{B} \to \mathbb{P}^\natural \dashrightarrow \mathbb{P}^\natural$.

The abstract domain $\mathbb{D}^\natural$ is an algebra while the domain of abstract properties $\mathbb{P}^\natural$ is a set. So the mathematical structures are different. However, following mathematicians that call $\mathbb{Z}$ the "ring of integers" where a ring is an algebraic structure and $\mathbb{Z}$ is a set, we often say, by abuse of language, that $\mathbb{P}^\natural$ an abstract domain.

en.wikipedia.org/wiki/Abstract_algebra

Abstract structural semantics/interpreter

The semantics can be implemented as instances of a generic abstract interpreter defined below.

- *Abstract semantics outside a statement* $\mathtt{S}$

$$\ell \notin \mathsf{labs}[\![\mathtt{S}]\!] \quad \Rightarrow \quad \widehat{\mathcal{S}}^{\,\natural}[\![\mathtt{S}]\!]\,\mathcal{R}_0\,\ell \triangleq \bot^{\natural} \tag{21.3}$$

- *Abstract semantics of a program* $\mathtt{P} ::= \mathtt{Sl}\ \ell'$

$$\widehat{\mathcal{S}}^{\,\natural}[\![\mathtt{P}]\!] \quad \triangleq \quad \widehat{\mathcal{S}}^{\,\natural}[\![\mathtt{Sl}]\!] \tag{21.4}$$

en.wikipedia.org/wiki/Interpreter_(computing)

- *Abstract semantics of a statement list* $\mathtt{sl} ::= \mathtt{sl}'\ \mathtt{s}$

$$\widehat{\mathcal{S}}^{\natural}[\![\mathtt{sl}]\!]\,\mathcal{R}_0\,\ell \quad \triangleq \quad \big(\!\big|\, \ell \in \mathsf{labs}[\![\mathtt{sl}']\!] \setminus \{\mathsf{at}[\![\mathtt{s}]\!]\} \;?\; \widehat{\mathcal{S}}^{\natural}[\![\mathtt{sl}']\!]\,\mathcal{R}_0\,\ell \;\big|\!\big| \;\big|\!\big|\, \ell \in \mathsf{labs}[\![\mathtt{s}]\!] \;?\; \widehat{\mathcal{S}}^{\natural}[\![\mathtt{s}]\!](\widehat{\mathcal{S}}^{\natural}[\![\mathtt{sl}']\!]\,\mathcal{R}_0\,\mathsf{at}[\![\mathtt{s}]\!])\,\ell \;:\; \bot^{\natural} \,\big)\!\big| \tag{21.5}$$

- *Abstract semantics of an empty statement list* $\mathtt{sl} ::= \epsilon$

$$\widehat{\mathcal{S}}^{\natural}[\![\mathtt{sl}]\!]\,\mathcal{R}_0\,\ell \quad \triangleq \quad \big(\!\big|\, \ell = \mathsf{at}[\![\mathtt{sl}]\!] \;?\; \mathcal{R}_0 \;:\; \bot^{\natural} \,\big)\!\big| \tag{21.6}$$

- *Abstract semantics of an assignment statement* $\mathtt{S ::= x = A ;}$

$$\widehat{\mathcal{S}}^{\natural}[\![\mathtt{S}]\!]\,\mathcal{R}_0\,\ell = \left(\!\!\left|\, \ell = \mathrm{at}[\![\mathtt{S}]\!] \;?\; \mathcal{R}_0 \;\middle|\; \ell = \mathrm{after}[\![\mathtt{S}]\!] \;?\; \mathrm{assign}_{\natural}[\![\mathtt{x},\mathtt{A}]\!]\,\mathcal{R}_0 \;:\; \bot^{\natural} \,\right|\!\!\right) \tag{21.7}$$

where $\mathrm{assign}[\![\mathtt{x},\mathtt{A}]\!] \circ \gamma \sqsubseteq \gamma \circ \mathrm{assign}_{\natural}[\![\mathtt{x},\mathtt{A}]\!]$.

- *Abstract semantics of a skip statement* $\mathtt{S ::= ;}$

$$\widehat{\mathcal{S}}^{\natural}[\![\mathtt{S}]\!]\,\mathcal{R}_0\,\ell = \left(\!\!\left|\, \ell \in \{\mathrm{at}[\![\mathtt{S}]\!], \mathrm{after}[\![\mathtt{S}]\!]\} \;?\; \mathcal{R}_0 \;:\; \bot^{\natural} \,\right|\!\!\right) \tag{21.8}$$

- *Abstract semantics of a conditional statement* $\mathtt{S} ::= \mathtt{if\ (B)\ S}_t$

$$\begin{array}{rcl}\widehat{\mathcal{S}}^{\natural}[\![\mathtt{S}]\!]\,\mathcal{R}_0\,\ell & = & (\!|\, \ell = \mathrm{at}[\![\mathtt{S}]\!] \,?\, \mathcal{R}_0 \\ & & [\!|\, \ell \in \mathrm{in}[\![\mathtt{S}_t]\!] \,?\, \widehat{\mathcal{S}}^{\natural}[\![\mathtt{S}_t]\!]\,(\mathrm{test}^{\natural}[\![\mathtt{B}]\!]\,\mathcal{R}_0)\,\ell \\ & & [\!|\, \ell = \mathrm{after}[\![\mathtt{S}]\!] \,?\, \\ & & \quad \widehat{\mathcal{S}}^{\natural}[\![\mathtt{S}_t]\!]\,(\mathrm{test}^{\natural}[\![\mathtt{B}]\!]\,\mathcal{R}_0)\,\ell \sqcup^{\natural} \overline{\mathrm{test}}^{\natural}[\![\mathtt{B}]\!]\,\mathcal{R}_0 \\ & & \,:\, \bot^{\natural}\ |\!) \end{array} \tag{21.9}$$

where $\mathrm{test}[\![\mathtt{B}]\!] \circ \gamma \sqsubseteq \gamma \circ \mathrm{test}^{\natural}[\![\mathtt{B}]\!]$ and $\overline{\mathrm{test}}[\![\mathtt{B}]\!] \circ \gamma \sqsubseteq \gamma \circ \overline{\mathrm{test}}^{\natural}[\![\mathtt{B}]\!]$.

- *Abstract semantics of a conditional statement* $\mathtt{S} ::= \mathtt{if\ (B)\ } \mathtt{S}_t \mathtt{\ else\ } \mathtt{S}_f$

$$\begin{array}{rcl}
\widehat{\mathcal{S}}^{\natural}[\![\mathtt{S}]\!]\,\mathcal{R}_0\,\ell & = & (\!|\, \ell = \mathrm{at}[\![\mathtt{S}]\!] \mathbin{?} \mathcal{R}_0 \\
& & [\!]\ \ell \in \mathrm{in}[\![\mathtt{S}_t]\!] \mathbin{?} \widehat{\mathcal{S}}^{\natural}[\![\mathtt{S}_t]\!]\,(\mathrm{test}^{\natural}[\![\mathtt{B}]\!]\,\mathcal{R}_0)\,\ell \\
& & [\!]\ \ell \in \mathrm{in}[\![\mathtt{S}_f]\!] \mathbin{?} \widehat{\mathcal{S}}^{\natural}[\![\mathtt{S}_f]\!]\,(\overline{\mathrm{test}}^{\natural}[\![\mathtt{B}]\!]\,\mathcal{R}_0)\,\ell \\
& & [\!]\ \ell = \mathrm{after}[\![\mathtt{S}]\!] \mathbin{?} \\
& & \qquad \widehat{\mathcal{S}}^{\natural}[\![\mathtt{S}_t]\!]\,(\mathrm{test}^{\natural}[\![\mathtt{B}]\!]\,\mathcal{R}_0)\,\ell \sqcup^{\natural} \widehat{\mathcal{S}}^{\natural}[\![\mathtt{S}_f]\!]\,(\overline{\mathrm{test}}^{\natural}[\![\mathtt{B}]\!]\,\mathcal{R}_0)\,\ell \\
& & \mathbin{:} \bot^{\natural}\ |\!)
\end{array} \tag{21.10}$$

- *Abstract semantics of an iteration statement* $\mathtt{S} ::= \mathtt{while}^{\ell}\ \mathtt{(B)}\ \mathtt{S}_b$

$$\widehat{\mathcal{S}}^{\natural}[\![\mathtt{S}]\!]\,\mathcal{R}_0\,\ell' \quad = \quad \mathrm{lfp}^{\sqsubseteq^{\natural}}\,(\mathcal{F}^{\natural}[\![\mathtt{while}^{\ell}\ \mathtt{(B)}\ \mathtt{S}_b]\!]\,\mathcal{R}_0)\,\ell' \tag{21.11}$$

$$\mathcal{F}^{\natural}[\![\mathtt{while}^{\ell}\ \mathtt{(B)}\ \mathtt{S}_b]\!] \quad \in \quad \mathbb{P}^{\natural} \to ((\mathbb{L} \to \mathbb{P}^{\natural}) \to (\mathbb{L} \to \mathbb{P}^{\natural}))$$

$$\begin{array}{l}
\mathcal{F}^{\natural}[\![\mathtt{while}^{\ell}\ \mathtt{(B)}\ \mathtt{S}_b]\!]\,\mathcal{R}_0\,X\,\ell' \quad = \\
\quad (\!|\ \ell' = \ell\ ?\ \mathcal{R}_0 \sqcup^{\natural} \widehat{\mathcal{S}}^{\natural}[\![\mathtt{S}_b]\!]\,(\mathrm{test}^{\natural}[\![\mathtt{B}]\!]X(\ell))\,\ell \\
\quad\ |\!|\ \ell' \in \mathrm{in}[\![\mathtt{S}_b]\!] \setminus \{\ell\}\ ?\ \widehat{\mathcal{S}}^{\natural}[\![\mathtt{S}_b]\!]\,(\mathrm{test}^{\natural}[\![\mathtt{B}]\!]X(\ell))\,\ell' \\
\quad\ |\!|\ \ell' = \mathrm{after}[\![\mathtt{S}]\!]\ ?\ \overline{\mathrm{test}}^{\natural}[\![\mathtt{B}]\!]X(\ell) \sqcup^{\natural} \bigsqcup_{\ell'' \in \text{breaks-of}[\![\mathtt{S}_b]\!]}^{\natural} \widehat{\mathcal{S}}^{\natural}[\![\mathtt{S}_b]\!]\,(\mathrm{test}^{\natural}[\![\mathtt{B}]\!]X(\ell))\,\ell'' \\
\quad\ ;\ \bot^{\natural}\ |\!|)
\end{array}$$

- *Abstract invariant of an iteration statement* $\mathtt{S} ::= \mathtt{while}^{\ell}\ \mathtt{(B)}\ \mathtt{S}_b$

$$\widehat{\mathcal{S}}^{\natural}[\![\mathtt{S}]\!]\,\mathcal{R}_0\,\ell' = \mathsf{let}\ \overline{\mathcal{F}}^{\natural}[\![\mathtt{while}^{\ell}\ \mathtt{(B)}\ \mathtt{S}_b]\!] \in \mathbb{P}^{\natural} \xrightarrow{\nearrow} (\mathbb{P}^{\natural} \xrightarrow{\nearrow} \mathbb{P}^{\natural}) \tag{19.42}$$

$$\overline{\mathcal{F}}^{\natural}[\![\mathtt{while}^{\ell}\ \mathtt{(B)}\ \mathtt{S}_b]\!]\,\mathcal{R}_0\,X = \mathcal{R}_0 \cup \widehat{\mathcal{S}}^{\natural}[\![\mathtt{S}_b]\!]\,(\mathsf{test}^{\natural}[\![\mathtt{B}]\!]X)^{\ell}$$

$$\mathsf{and}\ I = \mathsf{lfp}^{\sqsubseteq^{\natural}}\ \overline{\mathcal{F}}^{\natural}[\![\mathtt{while}^{\ell}\ \mathtt{(B)}\ \mathtt{S}_b]\!]\,\mathcal{R}_0\ \mathsf{in}$$

$$\begin{array}{l}
\langle\!|\ \ell' = \ell\ ?\ I \\
\quad |\!|\ \ell' \in \mathsf{in}[\![\mathtt{S}_b]\!] \setminus \{\ell\}\ ?\ \widehat{\mathcal{S}}^{\natural}[\![\mathtt{S}_b]\!]\,(\mathsf{test}^{\natural}[\![\mathtt{B}]\!]\,I)\,\ell' \\
\quad |\!|\ \ell' = \mathsf{after}[\![\mathtt{S}]\!]\ ?\ \overline{\mathsf{test}}^{\natural}[\![\mathtt{B}]\!]\,I \sqcup^{\natural} \bigsqcup_{\ell'' \in \mathsf{breaks\text{-}of}[\![\mathtt{S}_b]\!]}^{\natural} \widehat{\mathcal{S}}^{\natural}[\![\mathtt{S}_b]\!]\,(\mathsf{test}^{\natural}[\![\mathtt{B}]\!]\,I)\,\ell'' \\
\quad :\ \bot^{\natural}\ |\!|\rangle
\end{array}$$

$I = (\mathsf{lfp}^{\sqsubseteq^{\natural}}\ \mathcal{F}^{\natural}[\![\mathtt{while}^{\ell}\ \mathtt{(B)}\ \mathtt{S}_b]\!]\,\mathcal{R}_0)^{\ell}$, see Exercise 19.18.

- *Abstract semantics of a break statement* $\mathtt{S} ::= \ell$ **`break ;`**

$$\widehat{\mathcal{S}}^{\natural}[\![\mathtt{S}]\!]\,\mathcal{R}_0\,\ell \quad = \quad (\!|\, \ell = \mathrm{at}[\![\mathtt{S}]\!] \;?\; \mathcal{R}_0 \;:\; \bot^{\natural} \,|\!) \tag{21.12}$$

- *Abstract semantics of a compound statement* $\mathtt{S} ::=$ `{ Sl }`

$$\widehat{\mathcal{S}}^{\natural}[\![\mathtt{S}]\!] \quad = \quad \widehat{\mathcal{S}}^{\natural}[\![\mathtt{Sl}]\!] \tag{21.13}$$

Example

For the program

$$\texttt{P} \quad = \quad \texttt{while}\ \ell_1\ \texttt{(x != 2)}\ \ell_2\ \texttt{break ;}\ \ell_3$$

with grammatical structure

$$\texttt{P}\left[\begin{array}{l} \texttt{Sl}_1\left[\begin{array}{l} \epsilon\left[\ \right] \\ \texttt{S}_2\left[\begin{array}{l} \texttt{while}\ \ell_1\ \texttt{(x != 2)} \\ \texttt{S}_3\left[\ \ell_2\ \texttt{break ;}\right.\end{array}\right.\end{array}\right. \\ \ell_3 \end{array}\right.$$

we have

$$\begin{aligned} \widehat{\mathcal{S}}^{\natural}[\![\texttt{P}]\!]\,\mathcal{R}_0\,\ell \quad = \quad & \big(\!\big|\ \ell = \ell_1\ ?\ \mathcal{R}_0 \ \big|\ \ell = \ell_2\ ?\ \texttt{test}^{\natural}[\![\texttt{x != 2}]\!]\mathcal{R}_0 \\ & \big|\ \ell = \ell_3\ ?\ \overline{\texttt{test}}^{\natural}[\![\texttt{x != 2}]\!]\mathcal{R}_0 \sqcup^{\natural} \texttt{test}^{\natural}[\![\texttt{x != 2}]\!]\mathcal{R}_0 \,:\, \bot^{\natural}\ \big|\!\big) \end{aligned}$$

The formal derivation is the following

$$
\begin{aligned}
& \widehat{\mathcal{S}}^{\natural}[\![\mathtt{P}]\!]\,\mathcal{R}_0\,\ell_3 \\
= \; & \widehat{\mathcal{S}}^{\natural}[\![\mathtt{Sl}_1]\!]\,\mathcal{R}_0\,\ell_{36} && \text{(21.4)} \\
= \; & \widehat{\mathcal{S}}^{\natural}[\![\mathtt{S}_2]\!]\,\mathcal{R}_0\,\ell_3 && \text{(21.5), labs}[\![\epsilon]\!] = \text{at}[\![\mathtt{S}_2]\!]\ell_1\text{, and (21.6)} \\
= \; & \mathrm{lfp}^{\sqsubseteq}(\mathcal{F}^{\natural}[\![\mathtt{while}\ \ell_1\ (\mathtt{x}\ !=\ 2)\ \mathtt{S}_3]\!]\,\mathcal{R}_0)\,\ell_3 && \text{(21.11)} \\
& \text{where } \mathcal{F}^{\natural}[\![\mathtt{while}\ \ell_1\ (\mathtt{x}\ !=\ 2)\ \ell_2\ \mathtt{break}\ ;\ \ell_3]\!]\,\mathcal{R}_0\,X\,\ell' \\
= \; & \big(\!\big|\ \ell' = \ell_1\ ?\ \mathcal{R}_0 \\
& \quad [\!]\ \ell' = \ell_2\ ?\ \mathrm{test}^{\natural}[\![\mathtt{x}\ !=\ 2]\!]X(\ell_1) \\
& \quad [\!]\ \ell' = \ell_3\ ?\ \overline{\mathrm{test}}^{\natural}[\![\mathtt{x}\ !=\ 2]\!]X(\ell_1) \sqcup^{\natural} \mathrm{test}^{\natural}[\![\mathtt{x}\ !=\ 2]\!]X(\ell_1) \\
& \quad :\ \bot^{\natural}\ |\!\big) \\
& \qquad \text{(21.11), breaks-of}[\![\mathtt{S}_3]\!] = \{\ell_2\}\text{, in}[\![\mathtt{S}_3]\!] = \{\ell_1, \ell_2\}\text{, } \widehat{\mathcal{S}}^{\natural}[\![\mathtt{S}_3]\!]\,(\mathrm{test}^{\natural}[\![\mathtt{x}\ !=\ 2]\!]X(\ell_1))\,\ell_1 = \bot^{\natural} \\
& \qquad\ \text{by (21.12), } \mathcal{R}_0 \sqcup^{\natural} \bot^{\natural} = \mathcal{R}_0\text{, } \widehat{\mathcal{S}}^{\natural}[\![\mathtt{S}_3]\!]\,(\mathrm{test}^{\natural}[\![\mathtt{x}\ !=\ 2]\!]X(\ell_1))\,\ell_2 = \mathrm{test}^{\natural}[\![\mathtt{x}\ !=\ 2]\!]X(\ell_1) \\
& \qquad\ \text{by (21.12), and after}[\![\mathtt{S}]\!] = \ell_3 \\
= \; & \big(\!\big|\ \ell' = \ell_1\ ?\ \mathcal{R}_0\ [\!]\ \ell' = \ell_2\ ?\ \mathrm{test}^{\natural}[\![\mathtt{x}\ !=\ 2]\!]\mathcal{R}_0 \\
& \quad [\!]\ \ell' = \ell_3\ ?\ \overline{\mathrm{test}}^{\natural}[\![\mathtt{x}\ !=\ 2]\!]\mathcal{R}_0 \sqcup^{\natural} \mathrm{test}^{\natural}[\![\mathtt{x}\ !=\ 2]\!]\mathcal{R}_0\ :\ \bot^{\natural}\ |\!\big) && \text{since } X(\ell_1) = \mathcal{R}_0
\end{aligned}
$$

$\mathscr{F}^{¤}[\![\mathsf{S}_2]\!]\,\mathcal{R}_0\,X\,\ell'$ does not depend upon X, so it is the unique hence least fixpoint. In general $\overline{\mathsf{test}}^{¤}[\![\mathtt{x\ !=\ 2}]\!]\mathcal{R}_0 \sqcup^{¤} \mathsf{test}^{¤}[\![\mathtt{x\ !=\ 2}]\!]\mathcal{R}_0 = \mathcal{R}_0$ but this might not be the case for imprecise abstract domains (*e.g.* $\mathsf{test}^{¤}[\![\mathsf{B}]\!]\mathcal{R}_0 = \top^{¤}$ is sound).

What is the scientific reasoning here?

- We *observe* that different semantics of the language have similar structures
  - We *formalize* the semantics in the abstract, under hypotheses on the abstract domain
  - We show that the different semantics of the language are mere instances of the abstract semantics

What are the advantages of this approach?

- We show that the abstract semantics is well-defined (and so are all its instances)
- We will later show that the abstraction of the abstract interpreter is an instance of the abstract interpreter (uniquely defined by the abstraction of the abstract domain)

This is the principle of abstraction in mathematics
and the principle of economy of concepts in mathematics

`en.wikipedia.org/wiki/Abstraction_(mathematics)`

Well-definedness of the abstract interpreter

Theorem (15.31) If $F \in L \xrightarrow{uc} L \xrightarrow{uc} L$ be an upper continuous function on a cpo $\langle L, \sqsubseteq, \bot \rangle$ then $R \in L \mapsto \mathrm{lfp}^{\sqsubseteq} F(R) \in L \xrightarrow{uc} L$ is continuous.

Theorem (21.16, Well-definedness of the abstract semantics) The abstract semantics $\widehat{\mathcal{S}}^{\natural}[\![\mathtt{S}]\!]$ for well-defined abstract domain by $\mathbb{D}^{\natural}$ of Definition 21.1 (assuming $\mathrm{assign}_{\natural}[\![\mathtt{x}, \mathtt{A}]\!]$, $\mathrm{test}^{\natural}[\![\mathtt{B}]\!]$, and $\overline{\mathrm{test}}^{\natural}[\![\mathtt{B}]\!]$ to be continuous) is well-defined and continuous for any program component S.

Proof By structural induction, using Definition 21.1 for basic cases and Scott's iterative fixpoint Theorem 15.26 for (19.16). The proof shows simultaneously that $\widehat{\mathcal{S}}^{\natural}[\![\mathtt{S}]\!]\,\mathcal{R}_0$ is continuous in $\mathcal{R}_0$ for any program component S. This follows from the fact that the composition of continuous functions is continuous by Exercise 15.25 and Theorem 15.31 for the iteration. □

Once again upper-bounds of increasing chains need only to exist for the fixpoint iterations (Theorem 15.21) and the continuity hypothesis can be dispensed by considering transfinite fixpoint iterations [P. Cousot and R. Cousot, 1979].

Corollary (21.17, Well-definedness of the reachability semantics) The structural forward reachability semantics of Section **19.3** is well-defined.

Proof The reachability semantics $\mathcal{S}^{\vec{r}}[\![\mathsf{s}]\!]$ and $\mathcal{S}^{\vec{\mathsf{R}}}[\![\mathsf{s}]\!]$ and their common definition $\mathcal{S}^{\vec{\varrho}}[\![\mathsf{s}]\!]$ are easily checked to be an instance of the abstract interpreter and so by Theorem 21.16 are well-defined. □

Finitary abstract domains

An abstract domain $\mathbb{P}^\natural$ is *finitary* when it satisfies the *increasing chain condition* *i.e.* any strictly $\sqsubset^\natural$-increasing chain is finite. A finite abstract domain is finitary.

Theorem The abstract interpreter (*i.e.* a program implementing the specification $\widehat{\mathcal{S}}^\natural$) is well-defined (*i.e.* terminating) for finitary abstract domains.

Proof By Theorem 21.16, $\widehat{\mathcal{S}}^\natural$ is mathematically well-defined. By hypothesis the elements of $\mathbb{P}^\natural$ are computer representable. But fixpoint iterations might be infinite (*e.g.* for the reachability semantics of a non-terminating program). However, the abstract domain being finitary the least fixpoint can be computed by iteration from the infimum. □

Note that only the fixpoint iteration strictly increasing chains need to be finite (so the definition of finitary can be weakened to the iterates of fixpoints).

For mathematicians

- We would have defined the syntax as in Chapter **4**
- We would have specified the abstract domain and the structural abstract interpreter and proved their well-definedness (this Chapter **21**)
- We would have defined the prefix trace semantics of Chapter **17** as an instance of the abstract interpreter
- We would have proved that the abstraction of an instance of the abstract interpreter is an instance of the abstract interpreter (to be done in Chapter **27**)
- We would have defined the reachability abstract domain as an abstraction of the prefix trace abstract domain

  ... so getting Chapters **19** and **20** for free

  But most computer scientists expect concrete examples before abstract theorems

  And, historically, *e.g.* most groups considered in the first stage of the development of group theory were "concrete", having been realized through numbers, permutations, or matrices before the presentation by generators and relations.

  `en.wikipedia.org/wiki/History_of_group_theory`

Conclusion

- The idea of abstract interpreter is from [Kildall, 1973; Naur, 1965; Sintzoff, 1972; Wegbreit, 1975]
- The original idea of abstract interpreter was to execute the program on "non-standard" (*i.e.* abstract) finitary values (like signs or parity or constant + $\top/\bot$).
- This point of view is very restrictive (*e.g.* semantics do not work that way!)
- Generalization by [P. Cousot and R. Cousot, 1976, 1977] from (abstract) values to (abstract) properties
- [P. Cousot and R. Cousot, 1976] was the first to introduce infinitary abstract domains and a notion of soundness with respect to an operational semantics.
- Abstract domains and abstract interpreter are available online such as *e.g.* INTERPROC [Jeannet, 2009; Jeannet and Miné, 2009], PARMA POLYHEDRA LIBRARY (PPL) [Bagnara, Hill, and Zaffanella, 2008], EL/NA [Singh, Püschel, and Vechev, 2017]

Bibliography I

Bagnara, Roberto, Patricia M. Hill, and Enea Zaffanella (2008). "The Parma Polyhedra Library: Toward a complete set of numerical abstractions for the analysis and verification of hardware and software systems". *Sci. Comput. Program.* 72.1-2, pp. 3–21.

Cousot, Patrick and Radhia Cousot (1976). "Static determination of dynamic properties of programs". In: *Proceedings of the Second International Symposium on Programming*. Dunod, Paris, France, pp. 106–130.

— (1977). "Abstract Interpretation: A Unified Lattice Model for Static Analysis of Programs by Construction or Approximation of Fixpoints". In: *POPL*. ACM, pp. 238–252.

— (1979). "Constructive Versions of Tarski's Fixed Point Theorems". *Pacific Journal of Mathematics* 82.1, pp. 43–57.

Bibliography II

Jeannet, Bertrand (2009). "A web interface to the INTERPROC static analyzer". URL: http://pop-art.inrialpes.fr/interproc/interprocweb.cgi.

Jeannet, Bertrand and Antoine Miné (2009). "Apron: A Library of Numerical Abstract Domains for Static Analysis". In: *CAV*. Vol. 5643. Lecture Notes in Computer Science. Springer, pp. 661–667.

Kildall, Gary A. (1973). "A Unified Approach to Global Program Optimization". In: *POPL*. ACM Press, pp. 194–206.

Naur, Peter (Sept. 1965). "Checking of operand types in ALGOL compilers". *BIT Numerical Mathematics* 5, pp. 151–163.

Singh, Gagandeep, Markus Püschel, and Martin T. Vechev (2017). "Fast polyhedra abstract domain". In: *POPL*. ACM, pp. 46–59.

Bibliography III

Sintzoff, Michel (1972). "Calculating Properties of Programs by Valuations on Specific Models". In: *Proceedings of ACM Conference on Proving Assertions About Programs*. ACM, pp. 203–207.

Wegbreit, Ben (1975). "Property Extraction in Well-Founded Property Sets". *IEEE Trans. Software Eng.* 1.3, pp. 270–285.

Home work

- Read Ch. **21** "Abstract domain and abstract structural semantics" of

Principles of Abstract Interpretation
Patrick Cousot
MIT Press

The End, Thank you