Principles of Abstract Interpretation
MIT press

Ch. **44**, Software model checking

Patrick Cousot

pcousot.github.io

PrAbsInt@gmail.com github.com/PrAbsInt/

These slides are available at
http://github.com/PrAbsInt/slides/slides-44--model-checking-PrAbsInt.pdf

Chapter 44

Ch. **44**, Software model checking (1/3)

We have split our review of Chapter **44** into three videos

This first video is about

- regular specifications

Introduction

Objectives

- An introduction to software model-checking
- Designed by abstract interpretation, by calculational design of an abstraction of the semantics
- Using a non-conventional temporal specification (regular expressions instead of LTL, CTL, CTL*)

Specification of program semantics

Regular specifications

- We specify execution traces using regular expressions where terminals/[meta]-characters are replaced by local assertions
- A local assertion $\mathtt{L : B}$ specifies that invariant $\mathtt{B}$ should be true whenever execution reaches a program label $\ell \in \mathtt{L}$ in the set $\mathtt{L}$.
- $\mathtt{B}$ depends on the initial value $\underline{\mathtt{x}}$ of the variables $\mathtt{x}$ and there current value $\mathtt{x}$ at ℓ
- Abbreviation: $\mathtt{? : B} \triangleq \mathbb{L}\ \mathtt{: B}$ means that $\mathtt{B}$ holds at any program label $\ell \in \mathbb{L}$

Examples of regular specifications

- $(\texttt{? : x >= 0})^*$ states that the value of $\texttt{x}$ is always positive or zero during program execution.
- $(\texttt{? : x >=}\ \underline{\texttt{x}})^*$ states that the value of $\texttt{x}$ is always greater than or equal to its initial value $\underline{\texttt{x}}$ during execution.
- $(\texttt{? : x >= 0})^* \bullet \ell : \texttt{x == 0} \bullet (\texttt{? : x < 0})^*$ states that
  - the value of $\texttt{x}$ should be positive or zero, and next
  - if program point ℓ is ever reached then $\texttt{x}$ should be 0, and next
  - if computations go on after program point ℓ then $\texttt{x}$ should be negative afterwards.
- In the literature: Fred Schneider's security monitors [Schneider, 2000] : monitor the actions of a program, checks the behavior of the program against a given safety specification (and initiate remedial actions)[1,2]

---

[1] use automata equivalent to regular expressions

[2] use actions instead of program labels

]

Syntax of regular expressions

$\mathsf{L} \in \wp(\mathbb{L})$	sets of program labels
$\mathsf{x}, \mathsf{y}, \ldots \in \mathbb{V}$	program variables
$\underline{\mathsf{x}}, \underline{\mathsf{y}}, \ldots \in \underline{\mathbb{V}}$	initial values of variables
$\mathsf{B} \in \mathbb{B}$	boolean expressions such that $\mathrm{vars}[\![\mathsf{B}]\!] \subseteq \mathbb{V} \cup \underline{\mathbb{V}}$

$$
\begin{array}{lll}
\mathsf{R} & \in & \mathbb{R} \\
\mathsf{R} & ::= & \varepsilon \\
& \mid & \mathsf{L} : \mathsf{B} \\
& \mid & \mathsf{R}_1\mathsf{R}_2 \qquad (\text{or } \mathsf{R}_1 \bullet \mathsf{R}_2) \\
& \mid & \mathsf{R}_1 \mid \mathsf{R}_2 \\
& \mid & \mathsf{R}_1^* \quad \mid \quad \mathsf{R}_1^+ \\
& \mid & (\mathsf{R}_1)
\end{array}
$$

regular expressions (44.2)

- empty
- invariant B at L
- concatenation
- alternative
- zero/one or more occurrences of R
- grouping

Subsets of regular expressions

- $\mathbb{R}_\varepsilon$ empty regular expressions
- $\mathbb{R}^+$ non-empty regular expressions (used for specifications since no execution is empty)
- $\mathbb{R}^{\nmid}$ alternative $\mid$-free regular expressions

Semantics of regular expressions

- The semantics $\mathcal{S}^{\mathsf{r}}[\![\mathsf{R}]\!]$ of a regular expression R is a relation between
  - an initial environment $\underline{\varrho}$ (holding the initial values of variables), and
  - the traces π from $\underline{\varrho}$ satisfying the regular specification R
- Example:
  - $\mathsf{R} \triangleq \ell : \mathsf{x} = \underline{\mathsf{x}} \bullet \ell' : \mathsf{x} = \underline{\mathsf{x}} + 1$
  - $\mathcal{S}^{\mathsf{r}}[\![\mathsf{R}]\!] = \{\langle \underline{\varrho}, \langle \ell_1, \rho\rangle\langle \ell_2, \rho'\rangle\rangle \mid \rho(\mathsf{x}) = \underline{\varrho}(\mathsf{x}) \wedge \rho'(\mathsf{x}) = \underline{\varrho}(\mathsf{x}) + 1\}[]$

  - The program ℓ_1 `x = x + 1 ;` ℓ_2 satisfies this specification
  - The program ℓ_1 `x = x + 1 ;` ℓ_2 `x = x + 1 ;` ℓ_3 also satisfies this specification (the execution can be longer than the specification)
  - The program ℓ_1 `y = 0 ;` ℓ_2 does not satisfy this specification

Semantics of regular expressions (Cont'd)

Semantics of boolean expressions

$$
\begin{aligned}
\mathscr{A}[\![\mathsf{1}]\!]\underline{\varrho}, \rho &\triangleq 1 \\
\mathscr{A}[\![\underline{\mathsf{x}}]\!]\underline{\varrho}, \rho &\triangleq \underline{\varrho}(\mathsf{x}) \\
\mathscr{A}[\![\mathsf{x}]\!]\underline{\varrho}, \rho &\triangleq \rho(\mathsf{x}) \\
\mathscr{A}[\![\mathsf{A}_1 - \mathsf{A}_2]\!]\underline{\varrho}, \rho &\triangleq \mathscr{A}[\![\mathsf{A}_1]\!]\underline{\varrho}, \rho - \mathscr{A}[\![\mathsf{A}_2]\!]\underline{\varrho}, \rho \\
\mathscr{B}[\![\mathsf{A}_1 < \mathsf{A}_2]\!]\underline{\varrho}, \rho &\triangleq \mathscr{A}[\![\mathsf{A}_1]\!]\underline{\varrho}, \rho < \mathscr{A}[\![\mathsf{A}_2]\!]\underline{\varrho}, \rho \\
\mathscr{B}[\![\mathsf{B}_1 \ \mathsf{nand} \ \mathsf{B}_2]\!]\underline{\varrho}, \rho &\triangleq \mathscr{B}[\![\mathsf{B}_1]\!]\underline{\varrho}, \rho \uparrow \mathscr{B}[\![\mathsf{B}_2]\!]\underline{\varrho}, \rho
\end{aligned}
\tag{44.6}
$$

Semantics of regular expressions (Cont'd)

Semantics of regular expressions

$$
\begin{array}{rcl}
\mathcal{S}^{\mathsf{r}}[\![\varepsilon]\!] & \triangleq & \{\langle \underline{\varrho},\, \ni\rangle \mid \underline{\varrho} \in \mathbb{E}\mathrm{v}\} \\
\mathcal{S}^{\mathsf{r}}[\![\mathsf{L} : \mathsf{B}]\!] & \triangleq & \{\langle \underline{\varrho},\, \langle \ell,\, \rho\rangle\rangle \mid \ell \in \mathsf{L} \wedge \mathcal{B}[\![\mathsf{B}]\!]\underline{\varrho}, \rho\} \\
\mathcal{S}^{\mathsf{r}}[\![\mathsf{R}_1\mathsf{R}_2]\!] & \triangleq & \mathcal{S}^{\mathsf{r}}[\![\mathsf{R}_1]\!] \odot \mathcal{S}^{\mathsf{r}}[\![\mathsf{R}_2]\!] \\
\mathcal{S} \odot \mathcal{S}' & \triangleq & \{\langle \underline{\varrho},\, \pi \cdot \pi'\rangle \mid \langle \underline{\varrho},\, \pi\rangle \in \mathcal{S} \wedge \langle \underline{\varrho},\, \pi'\rangle \in \mathcal{S}'\} \\
\mathcal{S}^{\mathsf{r}}[\![\mathsf{R}_1 \mid \mathsf{R}_2]\!] & \triangleq & \mathcal{S}^{\mathsf{r}}[\![\mathsf{R}_1]\!] \cup \mathcal{S}^{\mathsf{r}}[\![\mathsf{R}_2]\!]
\end{array}
\qquad
\begin{array}{rcl}
\mathcal{S}^{\mathsf{r}}[\![\mathsf{R}]\!]^1 & \triangleq & \mathcal{S}^{\mathsf{r}}[\![\mathsf{R}]\!] \\
\mathcal{S}^{\mathsf{r}}[\![\mathsf{R}]\!]^{n+1} & \triangleq & \mathcal{S}^{\mathsf{r}}[\![\mathsf{R}]\!]^n \odot \mathcal{S}^{\mathsf{r}}[\![\mathsf{R}]\!] \\
\mathcal{S}^{\mathsf{r}}[\![\mathsf{R}^*]\!] & \triangleq & \bigcup_{n\in\mathbb{N}} \mathcal{S}^{\mathsf{r}}[\![\mathsf{R}]\!]^n \\
\mathcal{S}^{\mathsf{r}}[\![\mathsf{R}^+]\!] & \triangleq & \bigcup_{n\in\mathbb{N}\setminus\{0\}} \mathcal{S}^{\mathsf{r}}[\![\mathsf{R}]\!]^n \\
\mathcal{S}^{\mathsf{r}}[\![(\mathsf{R})]\!] & \triangleq & \mathcal{S}^{\mathsf{r}}[\![\mathsf{R}]\!]
\end{array}
\tag{44.7}
$$

Semantic properties to be analyzed

Semantics property

- The semantics of program P satisfies the specification R (for some initial environment $\underline{\varrho}$)
- Traditionally denoted $\mathsf{P}, \underline{\varrho} \models \mathsf{R}$
- "satisfies" means the prefix trace semantics of P is included in that of the specification R (extended to be long enough)

Definition 2 (Model checking)

$$\mathsf{P}, \underline{\varrho} \models \mathsf{R} \quad \triangleq \quad (\{\underline{\varrho}\} \times \widehat{\mathcal{S}}^{\,*}_{\mathbb{S}}[\![\mathsf{P}]\!]) \quad \subseteq \quad \alpha_{\mathsf{prefix}}(\mathcal{S}^{\mathsf{r}}[\![\mathsf{R} \bullet (? : \mathtt{tt})^{*}]\!]) \qquad \square$$

where

$$\alpha_{\mathsf{prefix}}(\Pi) \quad \triangleq \quad \{\langle \underline{\varrho},\ \pi \rangle \mid \pi \in \mathbb{S}^{+} \wedge \exists \pi' \in \mathbb{S}^{*} \,.\, \langle \underline{\varrho},\ \pi \cdot \pi' \rangle \in \Pi\} \qquad \text{prefix closure.}$$

the regular specification R specifies only a prefix of the traces of program P

This concludes our definition of

- regular specifications

from Chapter **44** (Software model checking)

The End

Principles of Abstract Interpretation
MIT press

Ch. **44**, Software model checking

Patrick Cousot

pcousot.github.io

PrAbsInt@gmail.com   github.com/PrAbsInt/

These slides are available at
http://github.com/PrAbsInt/slides/slides-44--model-checking-PrAbsInt.pdf

Chapter 44

Ch. **44**, Software model checking (2/3)

In this second video, we study

- the model-checking abstraction

Abstraction

Model checking is an boolean abstraction of the program semantics

$$\alpha_{\underline{\varrho},\mathsf{R}}(\Pi) \triangleq (\{\underline{\varrho}\} \times \Pi) \subseteq \alpha_{\mathsf{prefix}}(\mathcal{S}^{\mathsf{r}}[\![\mathsf{R} \bullet (? : \mathtt{tt})^*]\!])$$

$$\mathsf{P}, \underline{\varrho} \models \mathsf{R} = \alpha_{\underline{\varrho},\mathsf{R}}(\widehat{\mathcal{S}}^{\,*}_{\mathbb{S}}[\![\mathsf{P}]\!])$$

$$\langle \wp(\mathbb{S}^{+}), \subseteq \rangle \xleftarrow[\alpha_{\underline{\varrho},\mathsf{R}}]{\gamma_{\underline{\varrho},\mathsf{R}}} \langle \mathbb{B}, \Leftarrow \rangle$$

A short digression on regular expressions

Equivalence of regular expressions

- There are several ways of writing the same regular expression (*e.g.* `a+` or `a(a*)`)
- Notion of equivalence

$$\mathsf{R}_1 \backsimeq \mathsf{R}_2 \quad \triangleq \quad (\mathcal{S}^{\mathsf{r}}[\![\mathsf{R}_1]\!] = \mathcal{S}^{\mathsf{r}}[\![\mathsf{R}_2]\!])$$

- Equivalent regular expressions have the same semantics

Disjunctive normal form of regular expressions

- A regular expression is in disjunctive normal form if it is of the form $(\mathtt{R}_1 \mid \ldots \mid \mathtt{R}_n)$ for some $n \geqslant 1$, in which none of the $\mathtt{R}_i$, for $1 \leqslant i \leqslant n$, contains an occurrence of $\mid$.
- Kleene's algorithm transforms any regular expression $\mathtt{R}$ into an equivalent disjunctive normal form $\mathrm{dnf}(\mathtt{R})$ (so $\mathrm{dnf}(\mathtt{R}) \doteqdot \mathtt{R}$)

$$
\begin{array}{rcl}
\mathrm{dnf}(\varepsilon) & \triangleq & \varepsilon \\
\mathrm{dnf}(\mathtt{R}_1 \mid \mathtt{R}_2) & \triangleq & \mathrm{dnf}(\mathtt{R}_1) \mid \mathrm{dnf}(\mathtt{R}_2) \\
\mathrm{dnf}(\mathtt{R}^*) & \triangleq & \mathrm{let}\ \mathtt{R}^1 \mid \ldots \mid \mathtt{R}^n = \mathrm{dnf}(\mathtt{R})\ \mathrm{in}\ ((\mathtt{R}^1)^* \ldots (\mathtt{R}^n)^*)^* \\
\mathrm{dnf}(\mathtt{R}_1\mathtt{R}_2) & \triangleq & \mathrm{let}\ \mathtt{R}_1^1 \mid \ldots \mid \mathtt{R}_1^{n_1} = \mathrm{dnf}(\mathtt{R}_1)\ \mathrm{and}\ \mathtt{R}_2^1 \mid \ldots \mid \mathtt{R}_2^{n_2} = \mathrm{dnf}(\mathtt{R}_2)\ \mathrm{in}\ \mathop{\Big|}_{i=1}^{n_1} \mathop{\Big|}_{j=1}^{n_2} \mathtt{R}_1^i \mathtt{R}_2^j
\end{array}
$$

$$
\begin{array}{rcl}
\mathrm{dnf}(\mathtt{L} : \mathtt{B}) & \triangleq & \mathtt{L} : \mathtt{B} \\
\mathrm{dnf}(\mathtt{R}^+) & \triangleq & \mathrm{dnf}(\mathtt{R}\mathtt{R}^*) \\
\mathrm{dnf}((\mathtt{R})) & \triangleq & (\mathrm{dnf}(\mathtt{R}))
\end{array}
$$

[Brzozowski, 1964] derivative of regular expressions

- a string of the form $a\sigma$ (starting with the symbol a) matches an expression $\mathtt{R}$ iff the suffix σ matches the *derivative* $D_a(\mathtt{R})$ (also denoted $a^{-1}\mathtt{R}$)
- Given a non-empty and alternative-free regular expression $\mathtt{R} \in \mathbb{R}^+ \cap \mathbb{R}^{\nmid}$, we define $\mathsf{fstnxt}(\mathtt{R}) = \langle \mathtt{L} : \mathtt{B},\ \mathtt{R}' \rangle$ such that
  - $\mathtt{L} : \mathtt{B}$ recognizes the first state of sequences of states recognized by $\mathtt{R}$;
  - the derivative $\mathtt{R}'$ recognizes sequences of states after the first state of sequences of states recognized by $\mathtt{R}$.

$$
\begin{array}{rcll}
\mathsf{fstnxt}(\mathtt{L} : \mathtt{B}) & \triangleq & \langle \mathtt{L} : \mathtt{B},\ \varepsilon \rangle & (44.18)\\
\mathsf{fstnxt}(\mathtt{R}_1\mathtt{R}_2) & \triangleq & \mathsf{fstnxt}(\mathtt{R}_2) & \text{if } \mathtt{R}_1 \in \mathbb{R}_\varepsilon\\
\mathsf{fstnxt}(\mathtt{R}_1\mathtt{R}_2) & \triangleq & \text{let } \langle \mathtt{R}_1^f,\ \mathtt{R}_1^n \rangle = \mathsf{fstnxt}(\mathtt{R}_1) \text{ in } [\![\mathtt{R}_1^n \in \mathbb{R}_\varepsilon \ ?\ \langle \mathtt{R}_1^f,\ \mathtt{R}_2 \rangle \ \text{:}\ \langle \mathtt{R}_1^f,\ \mathtt{R}_1^n \bullet \mathtt{R}_2 \rangle]\!] & \\
 & & & \text{if } \mathtt{R}_1 \notin \mathbb{R}_\varepsilon\\
\mathsf{fstnxt}(\mathtt{R}^+) & \triangleq & \text{let } \langle \mathtt{R}^f,\ \mathtt{R}^n \rangle = \mathsf{fstnxt}(\mathtt{R}) \text{ in } [\![\mathtt{R}^n \in \mathbb{R}_\varepsilon \ ?\ \langle \mathtt{R}^f,\ \mathtt{R}^* \rangle \ \text{:}\ \langle \mathtt{R}^f,\ \mathtt{R}^n \bullet \mathtt{R}^* \rangle]\!] & \\
\mathsf{fstnxt}((\mathtt{R})) & \triangleq & \mathsf{fstnxt}(\mathtt{R}) &
\end{array}
$$

Calculational design of the abstract interpreter (I)

Methodology

- Apply the abstraction function

$$\alpha_{\underline{\varrho},\mathsf{R}}(\Pi) \triangleq (\{\underline{\varrho}\} \times \Pi) \subseteq \alpha_{\mathsf{prefix}}(\mathcal{S}^{\mathsf{r}}[\![\mathsf{R} \bullet (? : \mathsf{tt})^*]\!]))$$

  to the semantics

$$\widehat{\mathcal{S}}^{\,*}_{\mathbb{S}}[\![\mathsf{S}]\!]$$

  of program components S

- by structural induction on the program components S

Methodology

- Problem: $\alpha_{\underline{\varrho},\mathsf{R}}(\widehat{\mathcal{S}}^{*}_{\mathbb{s}}[\![\mathsf{s}]\!])$ is not structurally inductive on s
- Counter-example:

$$\alpha_{\underline{\varrho},\mathsf{R}}(\widehat{\mathcal{S}}^{*}_{\mathbb{s}}[\![\mathsf{s}_1\,;\mathsf{s}_2]\!]) \quad = \quad f_{\underline{\varrho},\mathsf{R}}(\alpha_{\underline{\varrho},\mathsf{R}_1}(\widehat{\mathcal{S}}^{*}_{\mathbb{s}}[\![\mathsf{s}_1]\!]), \alpha_{\underline{\varrho},\mathsf{R}_2}(\widehat{\mathcal{S}}^{*}_{\mathbb{s}}[\![\mathsf{s}_2]\!]))$$

where $\mathsf{R} = \mathsf{R}_1\mathsf{R}_2$, R_1 specifies s_1, and R_2 specifies s_2

How do we get R_1 and R_2???

- Solution: use a more refined abstraction
  - Checking that s satisfies the beginning R_1 of R
  - Returns the remaining R_2 of R at the end of s

$$\begin{aligned}\alpha_{\underline{\varrho},\mathsf{R}}(\widehat{\mathcal{S}}^{*}_{\mathbb{s}}[\![\mathsf{s}_1\,;\mathsf{s}_2]\!]) \quad = \quad & \text{let } \langle b_1,\ \mathsf{R}_2\rangle = \alpha_{\underline{\varrho},\mathsf{R}}(\widehat{\mathcal{S}}^{*}_{\mathbb{s}}[\![\mathsf{s}_1]\!]) \text{ in}\\ & \quad \text{let } \langle b_2,\ \mathsf{R}_3\rangle = \alpha_{\underline{\varrho},\mathsf{R}_2}(\widehat{\mathcal{S}}^{*}_{\mathbb{s}}[\![\mathsf{s}_2]\!])) \text{ in}\\ & \quad\quad \langle b_1 \wedge b_2,\ \mathsf{R}_3\rangle\end{aligned}$$

Structural regular model-checking abstraction

Definition 44.23 of regular model checking

- We first consider the case of $|$-free regular expressions
- Trace model checking abstraction ($\underline{\varrho} \in \mathbb{E}\mathrm{v}$ is an initial environment and $\mathsf{R} \in \mathbb{R}^{+} \cap \mathbb{R}^{\nmid}$ is a non-empty and $|$-free regular expression):

$$\mathcal{M}^{t}\langle\underline{\varrho},\ \varepsilon\rangle\pi \triangleq \langle \mathrm{tt},\ \varepsilon\rangle \tag{44.24}$$

$$\mathcal{M}^{t}\langle\underline{\varrho},\ \mathsf{R}\rangle\ni \triangleq \langle \mathrm{tt},\ \mathsf{R}\rangle$$

$$\mathcal{M}^{t}\langle\underline{\varrho},\ \mathsf{R}\rangle\pi \triangleq \text{let}\ \langle \ell_1,\ \rho_1\rangle\pi' = \pi\ \text{and}\ \langle \mathsf{L} : \mathsf{B},\ \mathsf{R}'\rangle = \mathrm{fstnxt}(\mathsf{R})\ \text{in} \qquad \pi \neq \ni$$

$$\llparenthesis\, \langle\underline{\varrho},\ \langle \ell_1,\ \rho_1\rangle\rangle \in \mathcal{S}^{\mathsf{r}}[\![\mathsf{L} : \mathsf{B}]\!]\ ?\ \mathcal{M}^{t}\langle\underline{\varrho},\ \mathsf{R}'\rangle\pi'\ \text{:}\ \langle \mathrm{ff},\ \mathsf{R}'\rangle\, \rrparenthesis$$

Example

- $\pi = \langle \ell_1,\ \rho_1 \rangle \pi'$ with $\pi' = \langle \ell_2,\ \rho_2 \rangle \ni$ with $\rho_2 = \rho_1[\mathsf{x} \leftarrow \rho_1(\mathsf{x}) + 1]$ is a trace of $\widehat{\mathcal{S}}^{\,*}_{\mathbb{S}}[\![\ell_1\ \mathbf{x = x + 1\ ;}\ell_2]\!]$
- $\mathsf{R}_1 = \mathbf{?}\ \mathbf{:}\ \mathsf{x} = \underline{\mathsf{x}} \bullet \mathbf{?}\ \mathbf{:}\ \mathsf{x} = \underline{\mathsf{x}} + 1 \bullet \mathbf{?}\ \mathbf{:}\ \mathsf{x} = \underline{\mathsf{x}} + 3$
- $\mathsf{fstnxt}(\mathsf{R}_1) = \langle \mathbb{L}\ \mathbf{:}\ \mathsf{x} = \underline{\mathsf{x}},\ \mathsf{R}_2 \rangle$ with $\mathsf{R}_2 = \mathbf{?}\ \mathbf{:}\ \mathsf{x} = \underline{\mathsf{x}} + 1 \bullet \mathbf{?}\ \mathbf{:}\ \mathsf{x} = \underline{\mathsf{x}} + 3$
- $\mathsf{fstnxt}(\mathsf{R}_2) = \langle \mathbb{L}\ \mathbf{:}\ \mathsf{x} = \underline{\mathsf{x}} + 1,\ \mathsf{R}_3 \rangle$ with $\mathsf{R}_3 = \mathbf{?}\ \mathbf{:}\ \mathsf{x} = \underline{\mathsf{x}} + 3$
- $\mathcal{M}^t\langle \underline{\varrho},\ \mathsf{R}_3 \rangle \ni = \langle \mathsf{tt},\ \varepsilon \rangle$
- $\langle \underline{\varrho},\ \langle \ell_2,\ \rho_2 \rangle \rangle \in \mathcal{S}^{\mathsf{r}}[\![\mathbb{L}\ \mathbf{:}\ \mathsf{x} = \underline{\mathsf{x}} + 1]\!] = \rho_2(\mathsf{x}) = \underline{\varrho}(\underline{\mathsf{x}}) + 1$
- $\mathcal{M}^t\langle \underline{\varrho},\ \mathsf{R}_2 \rangle \pi' = \llparenthesis\, \langle \underline{\varrho},\ \langle \ell_2,\ \rho_2 \rangle \rangle \in \mathcal{S}^{\mathsf{r}}[\![\mathbb{L}\ \mathbf{:}\ \mathsf{x} = \underline{\mathsf{x}} + 1]\!]\ ?\ \mathcal{M}^t\langle \underline{\varrho},\ \mathsf{R}_3 \rangle \ni\ :\ \langle \mathsf{ff},\ \mathsf{R}_3 \rangle \,\rrparenthesis = \llparenthesis\, \rho_2(\mathsf{x}) = \underline{\varrho}(\underline{\mathsf{x}}) + 1\ ?\ \langle \mathsf{tt},\ \varepsilon \rangle\ :\ \langle \mathsf{ff},\ \mathsf{R}_3 \rangle \,\rrparenthesis$
- $\langle \underline{\varrho},\ \langle \ell_1,\ \rho_1 \rangle \rangle \in \mathcal{S}^{\mathsf{r}}[\![\mathbb{L}\ \mathbf{:}\ \mathsf{x} = \underline{\mathsf{x}}]\!] = \rho_1(\mathsf{x}) = \underline{\varrho}(\underline{\mathsf{x}})$
- $\mathcal{M}^t\langle \underline{\varrho},\ \mathsf{R}_1 \rangle \pi \triangleq \llparenthesis\, \langle \underline{\varrho},\ \langle \ell_1,\ \rho_1 \rangle \rangle \in \mathcal{S}^{\mathsf{r}}[\![\mathbb{L}\ \mathbf{:}\ \mathsf{x} = \underline{\mathsf{x}}]\!]\ ?\ \mathcal{M}^t\langle \underline{\varrho},\ \mathsf{R}_2 \rangle \pi'\ :\ \langle \mathsf{ff},\ \mathsf{R}_2 \rangle \,\rrparenthesis = \llparenthesis\, \rho_1(\mathsf{x}) = \underline{\varrho}(\underline{\mathsf{x}})\ ?\ \mathcal{M}^t\langle \underline{\varrho},\ \mathsf{R}_2 \rangle \pi'\ :\ \langle \mathsf{ff},\ \mathsf{R}_2 \rangle \,\rrparenthesis = \llparenthesis\, \rho_1(\mathsf{x}) = \underline{\varrho}(\underline{\mathsf{x}})\ ?\ \llparenthesis\, \rho_2(\mathsf{x}) = \underline{\varrho}(\underline{\mathsf{x}}) + 1\ ?\ \langle \mathsf{tt},\ \varepsilon \rangle\ :\ \langle \mathsf{ff},\ \mathsf{R}_3 \rangle \,\rrparenthesis\ :\ \langle \mathsf{ff},\ \mathsf{R}_2 \rangle \,\rrparenthesis$ $\leftarrow$ if ff we could also return the counter-example π

Definition 44.23 of regular model checking (Cont'd)

- Set of traces model checking abstraction (for an $\mid$-free regular expression $\mathsf{R} \in \mathbb{R}^{+}$):

$$\mathcal{M}^{+}\langle\underline{\varrho},\ \mathsf{R}\rangle\Pi \triangleq \{\langle\pi,\ \mathsf{R}'\rangle \mid \pi \in \Pi \wedge \langle\mathtt{tt},\ \mathsf{R}'\rangle = \mathcal{M}^{t}\langle\underline{\varrho},\ \mathsf{R}\rangle\pi\} \tag{44.25}$$

This abstraction is a Galois connection

$$\langle\wp(\mathbb{S}^{+}),\ \subseteq\rangle \xleftarrow[\mathcal{M}^{+}\langle\underline{\varrho},\ \mathsf{R}\rangle]{\gamma_{\mathcal{M}^{+}\langle\underline{\varrho},\ \mathsf{R}\rangle}} \langle\wp(\mathbb{S}^{+} \times \mathbb{R}^{+}),\ \subseteq\rangle \quad \text{for } \mathsf{R} \in \mathbb{R}^{+} \text{ in (44.25)} \tag{44.30}$$

- Program component $\mathsf{S} \in \mathbb{P}\mathbb{c}$ model checking (for an $\mid$-free regular expression $\mathsf{R} \in \mathbb{R}^{+}$):

$$\mathcal{M}^{+}[\![\mathsf{S}]\!]\langle\underline{\varrho},\ \mathsf{R}\rangle \triangleq \mathcal{M}^{+}\langle\underline{\varrho},\ \mathsf{R}\rangle(\widehat{\mathcal{S}}^{\,*}_{\mathbb{S}}[\![\mathsf{S}]\!]) \tag{44.26}$$

Definition 44.23 of regular model checking (Cont'd)

- We now consider the general case by decomposition into $\mid$-free regular expressions
- Set of traces model checking (for regular expression $\mathsf{R} \in \mathbb{R}$):

$$\mathcal{M}\langle\underline{\varrho},\ \mathsf{R}\rangle\Pi \triangleq \text{let}\,(\mathsf{R}_1 \mid \ldots \mid \mathsf{R}_n) = \text{dnf}(\mathsf{R}) \text{ in} \qquad (44.27)$$

$$\bigcup_{i=1}^{n} \{\pi \mid \exists \mathsf{R}' \in \mathbb{R}\ .\ \langle\pi,\ \mathsf{R}'\rangle \in \mathcal{M}^{\dagger}\langle\underline{\varrho},\ \mathsf{R}_i\rangle\Pi\}$$

This abstraction is a Galois connection

$$\langle\wp(\mathbb{S}^+),\ \subseteq\rangle \xrightleftharpoons[\mathcal{M}\langle\underline{\varrho},\ \mathsf{R}\rangle]{\gamma_{\mathcal{M}\langle\underline{\varrho},\ \mathsf{R}\rangle}} \langle\wp(\mathbb{S}^+),\ \subseteq\rangle \quad \text{for } \mathsf{R} \in \mathbb{R} \text{ in } (44.27) \qquad (44.31)$$

- Model checking of a program component $\mathsf{S} \in \mathbb{P}\mathbb{c}$ (for regular expression $\mathsf{R} \in \mathbb{R}$):

$$\mathcal{M}[\![\mathsf{S}]\!]\langle\underline{\varrho},\ \mathsf{R}\rangle \triangleq \mathcal{M}\langle\underline{\varrho},\ \mathsf{R}\rangle(\widehat{\mathcal{S}}^{\,*}_{\mathbb{S}}[\![\mathsf{S}]\!]) \qquad (44.28)$$

Definition 44.23 of regular model checking (Cont'd)

- Back to boolean model-checking

$$\langle \wp(\mathbb{S}^{+}),\ \subseteq \rangle \xleftarrow[\alpha_{\mathcal{M}\langle \underline{\varrho},\,\mathsf{R}\rangle}]{\gamma_{\mathcal{M}\langle \underline{\varrho},\,\mathsf{R}\rangle}} \langle \mathbb{B},\ \Leftarrow \rangle \tag{44.32}$$

where $\alpha_{\mathcal{M}\langle \underline{\varrho},\,\mathsf{R}\rangle}(X) \quad \triangleq \quad (\{\underline{\varrho}\} \times X) \subseteq \mathcal{M}\langle \underline{\varrho},\ \mathsf{R}\rangle(X)$

Theorem 4 (Model checking soundness ($\Leftarrow$) and completeness ($\Rightarrow$))

$$\mathsf{P}, \underline{\varrho} \models \mathsf{R} \quad \Leftrightarrow \quad \alpha_{\mathcal{M}\langle \underline{\varrho},\,\mathsf{R}\rangle}(\widehat{\mathcal{S}}^{\,*}_{\mathbb{S}}[\![\mathsf{P}]\!])$$

□

Note that we can prove soundness/completeness from the specification of the model-checking algorithm (still to be designed)

Structural model checking

- We have solved the non-inductiveness problem!

Lemma 5 $\mathcal{M}^t\langle\underline{\varrho}, \mathsf{R}\rangle(\pi\frown\pi') = \langle \mathrm{tt}, \mathsf{R}'\rangle \Leftrightarrow (\exists \mathsf{R}'' \in \mathbb{R} \,.\, \mathcal{M}^t\langle\underline{\varrho}, \mathsf{R}\rangle(\pi) = \langle \mathrm{tt}, \mathsf{R}''\rangle \wedge \mathcal{M}^t\langle\underline{\varrho}, \mathsf{R}''\rangle(\pi') = \langle \mathrm{tt}, \mathsf{R}'\rangle)$. $\square$

- Structural model checking

$$\begin{cases} \widehat{\mathcal{M}}[\![\mathsf{S}]\!]\langle\underline{\varrho}, \mathsf{R}\rangle \triangleq \widehat{\mathcal{F}}[\![\mathsf{S}]\!](\prod_{\mathsf{S}' \triangleleft \mathsf{S}} \widehat{\mathcal{M}}[\![\mathsf{S}']\!])\langle\underline{\varrho}, \mathsf{R}\rangle \\ \mathsf{S} \in \mathbb{P}\mathbb{c} \end{cases}$$

The $\mathsf{S}' \triangleleft \mathsf{S}$ are the immediate components of program component S. By calculus,

Theorem 6 $\widehat{\mathcal{M}}[\![\mathsf{S}]\!]\langle\underline{\varrho}, \mathsf{R}\rangle = \mathcal{M}[\![\mathsf{S}]\!]\langle\underline{\varrho}, \mathsf{R}\rangle$. $\square$

This concludes our definition of

- the model-checking abstraction

from Chapter **44** (Software model checking)

The End

Principles of Abstract Interpretation

MIT press

Ch. **44**, Software model checking

Patrick Cousot

pcousot.github.io

PrAbsInt@gmail.com github.com/PrAbsInt/

These slides are available at
http://github.com/PrAbsInt/slides/slides-44--model-checking-PrAbsInt.pdf

Chapter 44

Ch. **44**, Software model checking (3/3)

In this third video, we study

- the calculational design of the structural model checker

Calculational design of the structural model-checking abstract interpreter (II)

Calculational design

$$
\begin{array}{ll}
& \boldsymbol{\mathscr{M}}[\![\mathsf{s}]\!]\langle\underline{\varrho},\ \mathsf{R}\rangle \\
\triangleq & \boldsymbol{\mathscr{M}}\langle\underline{\varrho},\ \mathsf{R}\rangle(\widehat{\boldsymbol{\mathscr{S}}}^{\,*}_{\mathbb{S}}[\![\mathsf{s}]\!]) \hfill \text{(44.28)} \\
= & \boldsymbol{\mathscr{M}}\langle\underline{\varrho},\ \mathsf{R}\rangle(\widehat{\mathscr{F}}_{\mathcal{S}}[\![\mathsf{s}]\!](\prod_{\mathsf{s}' \triangleleft \mathsf{s}} \widehat{\boldsymbol{\mathscr{S}}}^{\,*}_{\mathbb{S}}[\![\mathsf{s}']\!])\langle\underline{\varrho},\ \mathsf{R}\rangle)
\end{array}
$$

(by structural definition $\widehat{\boldsymbol{\mathscr{S}}}^{\,*}_{\mathbb{S}}[\![\mathsf{s}]\!] = \widehat{\mathscr{F}}_{\mathcal{S}}[\![\mathsf{s}]\!](\prod_{\mathsf{s}' \triangleleft \mathsf{s}} \widehat{\boldsymbol{\mathscr{S}}}^{\,*}_{\mathbb{S}}[\![\mathsf{s}']\!])$ of the stateful prefix trace semantics in Section **??**)

$= \ldots$ (calculus to expand definitions, rewrite and simplify formulæ by algebraic laws)

$$
= \widehat{\mathscr{F}}_{\mathcal{M}}[\![\mathsf{s}]\!](\prod_{\mathsf{s}' \triangleleft \mathsf{s}} \boldsymbol{\mathscr{M}}[\![\mathsf{s}']\!])\langle\underline{\varrho},\ \mathsf{R}\rangle
$$

(by calculational design to commute the model checking abstraction on the result to the model checking of the arguments of $\widehat{\boldsymbol{\mathscr{S}}}^{\,*}_{\mathbb{S}}[\![\mathsf{s}]\!]$)

$$
= \widehat{\mathscr{F}}_{\mathcal{M}}[\mathsf{s}](\prod_{\mathsf{s}' \triangleleft \mathsf{s}} \widehat{\boldsymbol{\mathscr{M}}}[\![\mathsf{s}']\!])\langle\underline{\varrho},\ \mathsf{R}\rangle \qquad \text{(ind. hyp.)}
$$

$$
\triangleq \widehat{\boldsymbol{\mathscr{M}}}[\![\mathsf{s}]\!]\langle\underline{\varrho},\ \mathsf{R}\rangle
$$

(by defining $\widehat{\boldsymbol{\mathscr{M}}}[\![\mathsf{s}]\!] \triangleq \widehat{\mathscr{F}}_{\mathcal{M}}[\![\mathsf{s}]\!](\prod_{\mathsf{s}' \triangleleft \mathsf{s}} \widehat{\boldsymbol{\mathscr{M}}}[\![\mathsf{s}']\!])$)

Calculational design

For iteration statements, $\widehat{\mathcal{T}}[\![\mathtt{S}]\!](\prod_{\mathtt{S}' \triangleleft \mathtt{S}} \widehat{\mathcal{S}}^{\,*}_{\mathbb{S}}[\![\mathtt{S}']\!])\langle \underline{\varrho}, \mathtt{R}\rangle$ is a fixpoint, and this proof involves the fixpoint transfer theorem [P. Cousot and R. Cousot, 1979, Th. 7.1.0.4 (3)] based on the commutation of the concrete and abstract transformer with the abstraction.

Theorem 7 (exact least fixpoint abstraction in a complete lattice) Assume that $\langle \mathcal{C}, \sqsubseteq, \bot, \top, \sqcup, \sqcap\rangle$ and $\langle \mathcal{A}, \preccurlyeq, 0, 1, \curlyvee, \curlywedge\rangle$ are complete lattices, $f \in \mathcal{C} \xrightarrow{\nearrow} \mathcal{C}$ is increasing, $\langle \mathcal{C}, \sqsubseteq\rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle \mathcal{A}, \preccurlyeq\rangle$, $\overline{f} \in \mathcal{A} \xrightarrow{\nearrow} \mathcal{A}$ is increasing, and $\alpha \circ f = \overline{f} \circ \alpha$ (*commutation property*). Then $\alpha(\mathsf{lfp}^{\sqsubseteq} f) = \mathsf{lfp}^{\preccurlyeq} \overline{f}$. $\square$

Structural regular model checking of an empty specification ε

$$\begin{array}{lll}
& \mathcal{M}^{+}[\![\mathsf{s}]\!]\langle\underline{\varrho},\ \varepsilon\rangle & \\
\triangleq & \mathcal{M}^{+}\langle\underline{\varrho},\ \varepsilon\rangle(\widehat{\mathcal{S}}^{*}_{\mathbb{S}}[\![\mathsf{s}]\!]) & \wr(44.26)\int \\
\triangleq & \{\langle\pi,\ \varepsilon'\rangle \mid \pi \in \widehat{\mathcal{S}}^{*}_{\mathbb{S}}[\![\mathsf{s}]\!] \wedge \langle\mathrm{tt},\ \varepsilon'\rangle = \mathcal{M}^{t}\langle\underline{\varrho},\ \varepsilon\rangle\pi\} & \wr(44.25)\int \\
\triangleq & \{\langle\pi,\ \varepsilon'\rangle \mid \pi \in \widehat{\mathcal{S}}^{*}_{\mathbb{S}}[\![\mathsf{s}]\!] \wedge \langle\mathrm{tt},\ \varepsilon'\rangle = \langle\mathrm{tt},\ \varepsilon\rangle\} & \wr\mathcal{M}^{t}\langle\underline{\varrho},\ \varepsilon\rangle\pi \triangleq \langle\mathrm{tt},\ \varepsilon\rangle \text{ by } (44.24)\int \\
= & \{\langle\pi,\ \varepsilon\rangle \mid \pi \in \widehat{\mathcal{S}}^{*}_{\mathbb{S}}[\![\mathsf{s}]\!]\} & \wr\text{def. } =\int \\
\triangleq & \widehat{\mathcal{M}}^{+}[\![\mathsf{s}]\!]\langle\underline{\varrho},\ \varepsilon\rangle &
\end{array}$$

Definition 44.39 (Structural model checking)

- Model checking an empty temporal specification ε.

$$\widehat{\mathcal{M}}^{+}[\![\mathsf{s}]\!]\langle\underline{\varrho},\ \varepsilon\rangle \triangleq \{\langle\pi,\ \varepsilon\rangle \mid \pi \in \widehat{\mathcal{S}}^{*}_{\mathbb{S}}[\![\mathsf{s}]\!]\} \qquad (44.41)$$

Structural regular model checking of programs $\mathtt{P} ::= \mathtt{Sl}$

$$\mathcal{M}[\![\mathtt{P}]\!]\langle\underline{\varrho},\ \mathtt{R}\rangle$$

$$\triangleq\ \mathcal{M}\langle\underline{\varrho},\ \mathtt{R}\rangle(\widehat{\mathcal{S}}^{\,*}_{\mathbb{S}}[\![\mathtt{P}]\!]) \qquad \text{\textlbrackdbl}(44.28)\text{\textrbrackdbl}$$

$$\triangleq\ \text{let}\,(\mathtt{R}_1 \mid \ldots \mid \mathtt{R}_n) = \text{dnf}(\mathtt{R})\ \text{in}\ \bigcup_{i=1}^{n}\{\pi \mid \exists \mathtt{R}' \in \mathbb{R}\ .\ \langle\pi,\ \mathtt{R}'\rangle \in \mathcal{M}^{+}\langle\underline{\varrho},\ \mathtt{R}_i\rangle(\widehat{\mathcal{S}}^{\,*}_{\mathbb{S}}[\![\mathtt{P}]\!])\} \qquad ((44.27))$$

$$=\ \text{let}\,(\mathtt{R}_1 \mid \ldots \mid \mathtt{R}_n) = \text{dnf}(\mathtt{R})\ \text{in}\ \bigcup_{i=1}^{n}\{\pi \mid \exists \mathtt{R}' \in \mathbb{R}\ .\ \langle\pi,\ \mathtt{R}'\rangle \in \mathcal{M}^{+}\langle\underline{\varrho},\ \mathtt{R}_i\rangle(\widehat{\mathcal{S}}^{\,*}_{\mathbb{S}}[\![\mathtt{Sl}]\!])\}$$

$$(\widehat{\mathcal{S}}^{\,*}_{\mathbb{S}}[\![\mathtt{P}]\!] = \widehat{\mathcal{S}}^{\,*}_{\mathbb{S}}[\![\mathtt{Sl}]\!])$$

$$=\ \text{let}\,(\mathtt{R}_1 \mid \ldots \mid \mathtt{R}_n) = \text{dnf}(\mathtt{R})\ \text{in}\ \bigcup_{i=1}^{n}\{\pi \mid \exists \mathtt{R}' \in \mathbb{R}\ .\ \langle\pi,\ \mathtt{R}'\rangle \in \widehat{\mathcal{M}}^{+}\langle\underline{\varrho},\ \mathtt{R}_i\rangle(\widehat{\mathcal{S}}^{\,*}_{\mathbb{S}}[\![\mathtt{Sl}]\!])\}$$

$$(\text{ind. hyp.})$$

$$=\ \text{let}\,(\mathtt{R}_1 \mid \ldots \mid \mathtt{R}_n) = \text{dnf}(\mathtt{R})\ \text{in}\ \bigcup_{i=1}^{n}\{\pi \mid \exists \mathtt{R}' \in \mathbb{R}\ .\ \langle\pi,\ \mathtt{R}'\rangle \in \widehat{\mathcal{M}}^{+}[\![\mathtt{Sl}]\!]\langle\underline{\varrho},\ \mathtt{R}_i\rangle\} \qquad ((44.26))$$

$$\triangleq\ \widehat{\mathcal{M}}[\![\mathtt{Sl}]\!]\langle\underline{\varrho},\ \mathtt{R}\rangle \qquad ((44.40))$$

Structural regular model checking of programs $\mathtt{P} ::= \mathtt{Sl}$ (Cont'd)

Definition 44.39 (Structural model checking, contn'd)

- Model checking a program $\mathtt{P} ::= \mathtt{Sl}\ \ell$ for a temporal specification $\mathtt{R} \in \mathbb{R}$ with alternatives.

$$\widehat{\mathcal{M}}[\![\mathtt{P}]\!]\langle \underline{\varrho}, \mathtt{R}\rangle \triangleq \mathrm{let}\,(\mathtt{R}_1 \mid \ldots \mid \mathtt{R}_n) = \mathrm{dnf}(\mathtt{R})\ \mathrm{in} \qquad (44.40)$$
$$\bigcup_{i=1}^{n} \{\pi \mid \exists \mathtt{R}' \in \mathbb{R} \,.\, \langle \pi, \mathtt{R}'\rangle \in \widehat{\mathcal{M}}^{+}[\![\mathtt{Sl}]\!]\langle \underline{\varrho}, \mathtt{R}_i\rangle\}$$

Structural regular model checking of assignments $\mathtt{S} ::= \ell\ \mathtt{x = A ;}$

Definition 44.39 (Structural model checking, contn'd)

- Model checking an assignment statement $\mathtt{S} ::= \ell\ \mathtt{x = A ;}$

$$
\begin{aligned}
\widehat{\mathscr{M}}^{\dagger}[\![\mathtt{S}]\!]\langle\underline{\varrho},\ \mathsf{R}\rangle \ \triangleq\ & \mathsf{let}\ \langle \mathsf{L} : \mathsf{B},\ \mathsf{R}'\rangle = \mathsf{fstnxt}(\mathsf{R})\ \mathsf{in} && (44.45)\\
& \{\langle\langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle,\ \mathsf{R}'\rangle \mid \langle\underline{\varrho},\ \langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle\rangle \in \mathscr{S}^{\mathsf{r}}[\![\mathsf{L} : \mathsf{B}]\!]\} && (a)\\
& \cup \{\langle\langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle\langle \mathsf{after}[\![\mathtt{S}]\!],\ \rho[\mathtt{x} \leftarrow \mathscr{A}[\![\mathtt{A}]\!]\rho]\rangle,\ \varepsilon\rangle \mid \mathsf{R}' \in \mathbb{R}_{\varepsilon} \wedge && (b)\\
& \qquad \langle\underline{\varrho},\ \langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle\rangle \in \mathscr{S}^{\mathsf{r}}[\![\mathsf{L} : \mathsf{B}]\!]\}\\
& \cup \{\langle\langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle\langle \mathsf{after}[\![\mathtt{S}]\!],\ \rho[\mathtt{x} \leftarrow \mathscr{A}[\![\mathtt{A}]\!]\rho]\rangle,\ \mathsf{R}''\rangle \mid \mathsf{R}' \notin \mathbb{R}_{\varepsilon} \wedge && (c)\\
& \qquad \langle\underline{\varrho},\ \langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle\rangle \in \mathscr{S}^{\mathsf{r}}[\![\mathsf{L} : \mathsf{B}]\!] \wedge \langle \mathsf{L}' : \mathsf{B}',\ \mathsf{R}''\rangle = \mathsf{fstnxt}(\mathsf{R}') \wedge\\
& \qquad \langle\underline{\varrho},\ \langle \mathsf{after}[\![\mathtt{S}]\!],\ \rho[\mathtt{x} \leftarrow \mathscr{A}[\![\mathtt{A}]\!]\rho]\rangle\rangle \in \mathscr{S}^{\mathsf{r}}[\![\mathsf{L}' : \mathsf{B}']\!]\}
\end{aligned}
$$

Structural regular model checking of assignments $\mathtt{S} ::= \ell\ \mathtt{x = A ;}$ (Cont'd)

$$\mathcal{M}^{\dagger}[\![\mathtt{S}]\!]\,\langle \underline{\varrho},\ \mathtt{R}\rangle$$

$$= \{\langle \pi,\ \mathtt{R}'\rangle \mid \pi \in \widehat{\mathcal{S}}^{\,*}_{\mathbb{S}}[\![\mathtt{Sl}]\!] \wedge \langle \mathrm{tt},\ \mathtt{R}'\rangle = \mathcal{M}^{t}\langle \underline{\varrho},\ \mathtt{R}\rangle\pi\} \qquad \text{\{(44.26) and (44.25)\}}$$

$$= \{\langle \pi,\ \mathtt{R}'\rangle \mid \pi \in \{\langle \ell,\ \rho\rangle \mid \rho \in \mathbb{E}\mathrm{v}\} \cup \{\langle \ell,\ \rho\rangle\langle \mathrm{after}[\![\mathtt{S}]\!],\ \rho[\mathtt{x} \leftarrow v]\rangle \mid \rho \in \mathbb{E}\mathrm{v} \wedge v = \mathcal{A}[\![\mathtt{A}]\!]\rho \wedge \langle \mathrm{tt},\ \mathtt{R}'\rangle = \mathcal{M}^{t}\langle \underline{\varrho},\ \mathtt{R}\rangle\pi\} \qquad \text{\{(42.4)\}}$$

$$= \{\langle\langle \ell,\ \rho\rangle,\ \mathtt{R}'\rangle \mid \rho \in \mathbb{E}\mathrm{v} \wedge \langle \mathrm{tt},\ \mathtt{R}'\rangle = \mathcal{M}^{t}\langle \underline{\varrho},\ \mathtt{R}\rangle\langle \ell,\ \rho\rangle\} \cup \{\langle\langle \ell,\ \rho\rangle\langle \mathrm{after}[\![\mathtt{S}]\!],\ \rho[\mathtt{x} \leftarrow v]\rangle,\ \mathtt{R}'\rangle \mid \rho \in \mathbb{E}\mathrm{v} \wedge v = \mathcal{A}[\![\mathtt{A}]\!]\rho \langle \mathrm{tt},\ \mathtt{R}'\rangle = \mathcal{M}^{t}\langle \underline{\varrho},\ \mathtt{R}\rangle\langle \ell,\ \rho\rangle\langle \mathrm{after}[\![\mathtt{S}]\!],\ \rho[\mathtt{x} \leftarrow v]\rangle\} \qquad \text{\{def. } \cup \text{ and } \in\text{\}}$$

$$= \{\langle\langle \ell,\ \rho\rangle,\ \mathtt{R}'\rangle \mid \langle \mathrm{tt},\ \mathtt{R}'\rangle = \mathrm{let}\ \langle \mathtt{L : B},\ \mathtt{R}''\rangle = \mathrm{fstnxt}(\mathtt{R})\ \mathrm{in}\ (\!|\ \langle \underline{\varrho},\ \langle \ell,\ \rho\rangle\rangle \in \mathcal{S}^{\mathrm{r}}[\![\mathtt{L : B}]\!]\ ?\ \langle \mathrm{tt},\ \mathtt{R}''\rangle : \langle \mathrm{ff},\ \mathtt{R}'\rangle\ |\!)\} \cup \{\langle\langle \ell,\ \rho\rangle\langle \mathrm{after}[\![\mathtt{S}]\!],\ \rho[\mathtt{x} \leftarrow v]\rangle,\ \mathtt{R}'\rangle \mid v = \mathcal{A}[\![\mathtt{A}]\!]\rho \wedge \langle \mathrm{tt},\ \mathtt{R}'\rangle = \mathrm{let}\ \langle \mathtt{L : B},\ \mathtt{R}''\rangle = \mathrm{fstnxt}(\mathtt{R})\ \mathrm{in}\ (\!|\ \langle \underline{\varrho},\ \langle \ell,\ \rho\rangle\rangle \in \mathcal{S}^{\mathrm{r}}[\![\mathtt{L : B}]\!]\ ?\ \mathcal{M}^{t}\langle \underline{\varrho},\ \mathtt{R}''\rangle\langle \mathrm{after}[\![\mathtt{S}]\!],\ \rho[\mathtt{x} \leftarrow v]\rangle : \langle \mathrm{ff},\ \mathtt{R}''\rangle\ |\!)\} \qquad \text{\{(44.24)\}}$$

$$= \ldots$$

$$
\begin{aligned}
&= \{\langle\langle \ell,\ \rho\rangle,\ \mathsf{R}'\rangle \mid \langle \mathsf{L} : \mathsf{B},\ \mathsf{R}'\rangle = \mathsf{fstnxt}(\mathsf{R}) \wedge \langle \underline{\varrho},\ \langle \ell,\ \rho\rangle\rangle \in \mathcal{S}^{\mathsf{r}}[\![\mathsf{L} : \mathsf{B}]\!]\} \cup \\
&\quad \{\langle\langle \ell,\ \rho\rangle\langle \mathsf{after}[\![\mathsf{S}]\!],\ \rho[\mathsf{x} \leftarrow v]\rangle,\ \mathsf{R}'\rangle \mid v = \mathcal{A}[\![\mathsf{A}]\!]\rho \wedge \exists \mathsf{R}'' \in \mathbb{R}\ .\ \langle \mathsf{L} : \mathsf{B},\ \mathsf{R}''\rangle = \mathsf{fstnxt}(\mathsf{R}) \wedge \langle \underline{\varrho},\ \langle \ell, \\
&\quad \rho\rangle\rangle \in \mathcal{S}^{\mathsf{r}}[\![\mathsf{L} : \mathsf{B}]\!] \wedge (\!|\, \mathsf{R}'' \in \mathbb{R}_\varepsilon\ ?\ \mathsf{tt} \ \text{\textsubdot{:}}\ \mathcal{M}^t\langle \underline{\varrho},\ \mathsf{R}''\rangle\langle \mathsf{after}[\![\mathsf{S}]\!],\ \rho[\mathsf{x} \leftarrow v]\rangle = \langle \mathsf{tt},\ \mathsf{R}'\rangle \,|\!)\} \\
&\hspace{20em} \text{(def. $=$ and $\mathcal{M}^t\langle \underline{\varrho},\ \varepsilon\rangle\pi \triangleq \langle \mathsf{tt},\ \varepsilon\rangle$ by (44.24))} \\
&= \{\langle\langle \ell,\ \rho\rangle,\ \mathsf{R}'\rangle \mid \langle \mathsf{L} : \mathsf{B},\ \mathsf{R}'\rangle = \mathsf{fstnxt}(\mathsf{R}) \wedge \langle \underline{\varrho},\ \langle \ell,\ \rho\rangle\rangle \in \mathcal{S}^{\mathsf{r}}[\![\mathsf{L} : \mathsf{B}]\!]\} \cup \\
&\quad \{\langle\langle \ell,\ \rho\rangle\langle \mathsf{after}[\![\mathsf{S}]\!],\ \rho[\mathsf{x} \leftarrow v]\rangle,\ \mathsf{R}'\rangle \mid v = \mathcal{A}[\![\mathsf{A}]\!]\rho \wedge \exists \mathsf{R}'' \in \mathbb{R}\ .\ \langle \mathsf{L} : \mathsf{B},\ \mathsf{R}''\rangle = \mathsf{fstnxt}(\mathsf{R}) \wedge \langle \underline{\varrho}, \\
&\quad \langle \ell,\ \rho\rangle\rangle \in \mathcal{S}^{\mathsf{r}}[\![\mathsf{L} : \mathsf{B}]\!] \wedge (\!|\, \mathsf{R}'' \in \mathbb{R}_\varepsilon\ ?\ \mathsf{tt}\ \text{\textsubdot{:}}\ \mathsf{let}\ \langle \mathsf{L}' : \mathsf{B}',\ \mathsf{R}'''\rangle = \mathsf{fstnxt}(\mathsf{R}'')\ \mathsf{in}\ \langle \underline{\varrho},\ \langle \mathsf{after}[\![\mathsf{S}]\!], \\
&\quad \rho[\mathsf{x} \leftarrow v]\rangle\rangle \in \mathcal{S}^{\mathsf{r}}[\![\mathsf{L}' : \mathsf{B}']\!] \,|\!)\} \hspace{20em} \text{((44.24))} \\
&= \mathsf{let}\ \langle \mathsf{L} : \mathsf{B},\ \mathsf{R}'\rangle = \mathsf{fstnxt}(\mathsf{R})\ \mathsf{in} \\
&\qquad \{\langle\langle \ell,\ \rho\rangle,\ \mathsf{R}'\rangle \mid \langle \underline{\varrho},\ \langle \ell,\ \rho\rangle\rangle \in \mathcal{S}^{\mathsf{r}}[\![\mathsf{L} : \mathsf{B}]\!]\} \\
&\quad \cup \{\langle\langle \ell,\ \rho\rangle\langle \mathsf{after}[\![\mathsf{S}]\!],\ \rho[\mathsf{x} \leftarrow v]\rangle,\ \varepsilon\rangle \mid v = \mathcal{A}[\![\mathsf{A}]\!]\rho \wedge \langle \underline{\varrho},\ \langle \ell,\ \rho\rangle\rangle \in \mathcal{S}^{\mathsf{r}}[\![\mathsf{L} : \mathsf{B}]\!] \wedge \mathsf{R}' \in \mathbb{R}_\varepsilon\} \\
&\quad \cup \{\langle\langle \ell,\ \rho\rangle\langle \mathsf{after}[\![\mathsf{S}]\!],\ \rho[\mathsf{x} \leftarrow v]\rangle,\ \mathsf{R}''\rangle \mid v = \mathcal{A}[\![\mathsf{A}]\!]\rho \wedge \langle \underline{\varrho},\ \langle \ell,\ \rho\rangle\rangle \in \mathcal{S}^{\mathsf{r}}[\![\mathsf{L} : \mathsf{B}]\!] \wedge \mathsf{R}' \notin \mathbb{R}_\varepsilon \wedge \mathsf{let}\ \langle \mathsf{L}' : \mathsf{B}', \\
&\quad \mathsf{R}''\rangle = \mathsf{fstnxt}(\mathsf{R}')\ \mathsf{in}\ \langle \underline{\varrho},\ \langle \mathsf{after}[\![\mathsf{S}]\!],\ \rho[\mathsf{x} \leftarrow v]\rangle\rangle \in \mathcal{S}^{\mathsf{r}}[\![\mathsf{L}' : \mathsf{B}']\!]\} \hspace{10em} \text{(def. $\cup$)} \\
&= \widehat{\mathcal{M}}^{+}[\![\mathsf{S}]\!]\ \langle \underline{\varrho},\ \mathsf{R}\rangle \hspace{20em} \text{((44.45))} \quad \square
\end{aligned}
$$

Structural regular model checking of a statement list $\texttt{Sl} ::= \texttt{Sl}'\ \texttt{S}$

Definition 44.39 (Structural model checking, contn'd)

- Model checking a statement list $\texttt{Sl} ::= \texttt{Sl}'\ \texttt{S}$

$$
\begin{aligned}
\widehat{\mathcal{M}}^{+}[\![\texttt{Sl}]\!]\langle \underline{\varrho}, \texttt{R}\rangle \quad \triangleq \quad & \widehat{\mathcal{M}}^{+}[\![\texttt{Sl}']\!]\langle \underline{\varrho}, \texttt{R}\rangle \\
& \cup \{\langle \pi \cdot \langle \texttt{at}[\![\texttt{S}]\!], \rho\rangle \cdot \pi', \texttt{R}''\rangle \mid \langle \pi \cdot \langle \texttt{at}[\![\texttt{S}]\!], \rho\rangle, \texttt{R}'\rangle \in \widehat{\mathcal{M}}^{+}[\![\texttt{Sl}']\!]\langle \underline{\varrho}, \texttt{R}\rangle \wedge \\
& \qquad \langle\langle \texttt{at}[\![\texttt{S}]\!], \rho\rangle \cdot \pi', \texttt{R}''\rangle \in \widehat{\mathcal{M}}^{+}[\![\texttt{S}]\!]\langle \underline{\varrho}, \texttt{R}'\rangle\}
\end{aligned}
\tag{44.42}
$$

Structural regular model checking of iterations $\mathtt{S} ::= \mathtt{while}\ ^{\ell}\ \mathtt{(B)}\ \mathtt{S}_b$

Definition 44.39 (Structural model checking, contn'd)

- Model checking an iteration statement $\mathtt{S} ::= \mathtt{while}\ ^{\ell}\ \mathtt{(B)}\ \mathtt{S}_b$

$$\widehat{\mathscr{M}}^{\dagger}[\![\mathtt{S}]\!]\langle\underline{\varrho},\ \mathtt{R}\rangle \triangleq \mathsf{lfp}^{\subseteq}\,(\widehat{\mathscr{F}}^{\dagger}[\![\mathtt{S}]\!]\langle\underline{\varrho},\ \mathtt{R}\rangle) \tag{44.49}$$

$$\widehat{\mathscr{F}}^{\dagger}[\![\mathtt{S}]\!]\langle\underline{\varrho},\ \mathtt{R}\rangle\ X \triangleq \ldots\ldots\ldots$$

Scalability

Convergence

- In practice, the set $\mathbb{S}$ of states must be assumed to be finite (and very small) and encoded symbolically
- Regular expressions may be replaced by finite automata
- Nevertheless, model-checking in general, and regular model checking in particular, does not scale
- Convergence acceleration methods (widening, narrowing, and duals) must be used (trivial example: bounded model checking limits the length of traces to an arbitrary length n)

Liveness

Liveness

- If the set of states is finite, this is safety
- Otherwise, abstraction is needed, BUT liveness is not preserved by over-approximation and under-approximation is difficult in infinite systems
- In general liveness in the finite abstract homomorphic transition does NOT imply liveness in the infinite concrete transition system, and
- non-liveness in the infinite concrete transition system does NOT imply non-liveness in the finite abstract transition system
- Our solution: variant functions.

Conclusion

Conclusion

- We have shown that a model-checker is an abstract interpretation of a program semantics [P. Cousot and R. Cousot, 2000]
- So the model-checker can be formally constructed by calculational design
- This provides a machine checkable [Jourdan, Laporte, Blazy, Leroy, and Pichardie, 2015] formal proof of soundness (and completeness) of the model-checker
- Soundness does not seem to be a preoccupation of the model-checking community!
- A computation tool (better than LaTeX editing, `grep`, and copy-paste) would be very helpful
- Pave the way for further non trivial abstractions (beyond the homomorphic abstractions)

Bibliography

References I

Brzozowski, Janusz A. (1964). "Derivatives of Regular Expressions". *J. ACM* 11.4, pp. 481–494.

Cousot, Patrick and Radhia Cousot (1979). "Systematic Design of Program Analysis Frameworks". In: *POPL*. ACM Press, pp. 269–282.

— (2000). "Temporal Abstract Interpretation". In: *POPL*. ACM, pp. 12–25.

Jourdan, Jacques-Henri, Vincent Laporte, Sandrine Blazy, Xavier Leroy, and David Pichardie (2015). "A Formally-Verified C Static Analyzer". In: *POPL*. ACM, pp. 247–259.

Schneider, Fred B. (2000). "Enforceable security policies". *ACM Trans. Inf. Syst. Secur.* 3.1, pp. 30–50.

Home work

Read Ch. **44** “Software model checking” of

Principles of Abstract Interpretation
Patrick Cousot
MIT Press

The End, Thank you