ICTAC 2019

October 30 – November 4, 2019

Hammamet, Tunisia

Calculational Design of a Regular Model Checker by Abstract Interpretation

Patrick Cousot

New York University, Courant Institute of Mathematics, Computer Science

pcousot@cs.nyu.edu   cs.nyu.edu/~pcousot

Introduction

How to design a static analyzer by abstract interpretation [P. Cousot and R. Cousot, 1977, 1979]

- Define the syntax & semantics of the language
- Define the semantic properties to be analyzed
- Define an abstraction of this semantic properties into an abstract domain (machine representable subset of the semantic properties)
- Design the static analyzer by calculational design of the abstraction of the semantics

How to design a static analyzer by abstract interpretation [P. Cousot and R. Cousot, 1977, 1979]

- Define the syntax & semantics of the language
- Define the semantic properties to be analyzed
- Define an abstraction of this semantic properties into an abstract domain (machine representable subset of the semantic properties)
- Design the static analyzer by calculational design of the abstraction of the semantics

- This is illustrated by the design of a regular model checker

  ("regular" means that the program behaviors are specified using regular expressions [Wolper, 1983])

Syntax and trace semantics of programs

Syntax

x, y, …	∈	$\mathbb{V}$		variable ($\mathbb{V}$ not empty)
A	∈	$\mathbb{A}$	::= 1 | x | A_1 - A_2	arithmetic expression
B	∈	$\mathbb{B}$	::= A_1 < A_2 | B_1 nand B_2	boolean expression
S	∈	$\mathbb{S}$	::=	statement
			x = A ;	assignment
		|	;	skip
		|	if (B) S | if (B) S else S	conditionals
		|	while (B) S | break ;	iteration and break
		|	{ Sl }	compound statement
Sl ∈ $\mathbb{Sl}$	::=	Sl S | ϵ		statement list
P ∈ $\mathbb{P}$	::=	Sl		program
S ∈ $\mathbb{Pc}$	$\triangleq$	$\mathbb{S} \cup \mathbb{Sl} \cup \mathbb{P}$		program component

Program labelling

Unique labelling to designate (sets of) program points $\ell \in \mathbb{L}$:

$\text{at}[\![\texttt{S}]\!]$	the program point at which execution of `S` starts;
$\text{after}[\![\texttt{S}]\!]$	the program exit point after `S`, at which execution of `S` is supposed to normally terminate, if ever;
$\text{escape}[\![\texttt{S}]\!]$	a boolean indicating whether or not the program component `S` contains a **`break ;`** statement escaping out of that component `S`;
$\text{break-to}[\![\texttt{S}]\!]$	the program point at which execution of the program component `S` goes to when a **`break ;`** statement escapes out of that component `S`;
$\text{breaks-of}[\![\texttt{S}]\!]$	the set of labels of all **`break ;`** statements that can escape out of `S`

Prefix traces

- Program label: $\ell \in \mathbb{L}$ (locates next step to be executed in the program)
- Environment: $\rho \in \mathbb{E}\mathbb{v} \triangleq \mathbb{V} \rightarrow \mathbb{V}$ assigns values $\rho(\mathbf{x}) \in \mathbb{V}$ to variables $\mathbf{x} \in \mathbb{V}$.
- State: $\langle \ell,\ \rho \rangle \in \mathbb{S} \triangleq (\mathbb{L} \times \mathbb{E}\mathbb{v})$
- Trace: finite or infinite sequence $\pi \in \mathbb{T}^{+\infty}$ of states
- Example: $\langle \ell_1,\ \{\mathbf{x} \rightarrow 1\} \rangle \langle \ell_2,\ \{\mathbf{x} \rightarrow 2\} \rangle \langle \ell_4,\ \{\mathbf{x} \rightarrow 2\} \rangle$
- Trace concatenation: $\frown$

$$
\begin{array}{rcll}
\pi_1\sigma_1 \frown \sigma_2\pi_2 & & & \text{undefined if } \sigma_1 \neq \sigma_2 \\
\pi_1 \frown \sigma_2\pi_2 & \triangleq & \pi_1 & \text{if } \pi_1 \in \mathbb{S}^{\infty} \text{ is infinite} \\
\pi_1\sigma_1 \frown \sigma_1\pi_2 & \triangleq & \pi_1\sigma_1\pi_2 & \text{if } \pi_1 \in \mathbb{T}^{+} \text{ is finite}
\end{array}
$$

- In pattern matching, we sometimes need the empty trace $\ni$. For example if $\sigma\pi\sigma' = \sigma$ then $\pi = \ni$ and $\sigma = \sigma'$.

Structural definitions

- Our definitions (semantics, modeled checking, etc) are structural *i.e.*by induction on the grammatical program structure

$$\begin{cases} \widehat{\mathcal{D}}[\![\mathsf{s}]\!]\, X \triangleq \widehat{\mathcal{T}}[\![\mathsf{s}]\!](\prod_{\mathsf{s}' \lhd \mathsf{s}} \widehat{\mathcal{D}}[\![\mathsf{s}']\!])\, X \\ \mathsf{s} \in \mathbb{P}\mathbb{c} \end{cases}$$

- the transformer $\widehat{\mathcal{T}}$ uses the results of the immediate components $\mathsf{s}' \lhd \mathsf{s}$ and involves a fixpoint computation for iteration statements.

Prefix trace semantics

- A prefix trace describes the beginning of a computation
- Evaluation of an arithmetic expression

$$\begin{aligned} \mathscr{A}[\![1]\!]\rho &\triangleq 1 \\ \mathscr{A}[\![x]\!]\rho &\triangleq \rho(x) \\ \mathscr{A}[\![A_1 - A_2]\!]\rho &\triangleq \mathscr{A}[\![A_1]\!]\rho - \mathscr{A}[\![A_2]\!]\rho \end{aligned} \quad (2)$$

- Assignment $S ::= \ell\ x = A\ ;$ (where $\text{at}[\![S]\!] = \ell$)

$$\widehat{\mathcal{S}}^{\,*}_{\mathbb{S}}[\![S]\!] = \{\langle\ell,\ \rho\rangle \mid \rho \in \mathbb{E}\mathbb{v}\} \cup \{\langle\ell,\ \rho\rangle\langle\text{after}[\![S]\!],\ \rho[x \leftarrow \mathscr{A}[\![A]\!]\rho]\rangle \mid \rho \in \mathbb{E}\mathbb{v}\} \quad (1)$$

Prefix trace semantics (cont'd)

- Break statement $\mathtt{S} ::= \ell$ **break ;** (where $\mathsf{at}[\![\mathtt{S}]\!] = \ell$)

$$\mathcal{S}^*[\![\mathtt{S}]\!] \triangleq \{\langle \ell,\ \rho\rangle \mid \rho \in \mathbb{E}\mathrm{v}\} \cup \{\langle \ell,\ \rho\rangle\langle \mathsf{break\text{-}to}[\![\mathtt{S}]\!],\ \rho\rangle \mid \rho \in \mathbb{E}\mathrm{v}\} \qquad (3)$$

Prefix trace semantics (cont'd)

- Conditional statement $\texttt{S} ::= \texttt{if}\ \ell\ \texttt{(B)}\ \texttt{S}_t$ (where $\mathsf{at}[\![\texttt{S}]\!] = \ell$)

$$\widehat{\mathcal{S}}^*[\![\texttt{S}]\!] \triangleq \{\langle\ell,\ \rho\rangle \mid \rho \in \mathbb{E}\mathsf{v}\} \cup \{\langle\ell,\ \rho\rangle\langle\mathsf{after}[\![\texttt{S}]\!],\ \rho\rangle \mid \mathcal{B}[\![\texttt{B}]\!]\rho = \mathsf{ff}\} \cup \{\langle\ell,\ \rho\rangle\langle\mathsf{at}[\![\texttt{S}_t]\!],\ \rho\rangle\pi \mid \mathcal{B}[\![\texttt{B}]\!]\rho = \mathsf{tt} \wedge \langle\mathsf{at}[\![\texttt{S}_t]\!],\ \rho\rangle\pi \in \widehat{\mathcal{S}}^*[\![\texttt{S}_t]\!]\} \quad (4)$$

- If the conditional statement $\texttt{S}$ is inside an iteration statement, and $\texttt{S}_t$ has a break, the execution goes on at the $\mathsf{break\text{-}to}[\![\texttt{S}]\!]$ after the iteration.

Prefix trace semantics (cont'd)

- Statement list $\texttt{Sl} ::= \texttt{Sl}'\ \texttt{S}$ (where $\mathsf{at}[\![\texttt{S}]\!] = \mathsf{after}[\![\texttt{Sl}']\!]$)

$$\widehat{\mathcal{S}}^*[\![\texttt{Sl}]\!] \triangleq \widehat{\mathcal{S}}^*[\![\texttt{Sl}']\!] \cup \widehat{\mathcal{S}}^*[\![\texttt{Sl}']\!] \mathbin{\frown} \mathcal{S}^*[\![\texttt{S}]\!] \tag{5}$$

$$\mathcal{S} \mathbin{\frown} \mathcal{S}' \triangleq \{\pi \mathbin{\frown} \pi' \mid \pi \in \mathcal{S} \wedge \pi' \in \mathcal{S}' \wedge \pi \mathbin{\frown} \pi' \text{ is well-defined}\}$$

- $\pi' \in \widehat{\mathcal{S}}^*[\![\texttt{S}]\!]$ starts $\mathsf{at}[\![\texttt{S}]\!] = \mathsf{after}[\![\texttt{Sl}']\!]$ so, by def. $\frown$, the trace $\pi \in \widehat{\mathcal{S}}^*[\![\texttt{Sl}']\!]$ must terminate to be able to go on with $\texttt{S}$.

Prefix trace semantics (cont'd)

- Empty statement list $\texttt{Sl} ::= \epsilon$ (where $\mathsf{at}[\![\texttt{Sl}]\!] \triangleq \mathsf{after}[\![\texttt{Sl}]\!]$)

$$\mathcal{S}^*[\![\texttt{Sl}]\!] \triangleq \{\langle \mathsf{at}[\![\texttt{Sl}]\!],\, \rho\rangle \mid \rho \in \mathbb{E}\mathbb{v}\}$$

Prefix trace semantics (cont'd)

- Iteration statement $\texttt{S} ::= \texttt{while}\ \ell\ \texttt{(B)}\ \texttt{S}_b$ (where $\text{at}[\![\texttt{S}]\!] = \ell$)

$$\widehat{\mathcal{S}}^*_{\mathbb{S}}[\![\texttt{while}\ \ell\ \texttt{(B)}\ \texttt{S}_b]\!] \quad = \quad \text{lfp}^{\subseteq}\, \mathcal{F}^*[\![\texttt{while}\ \ell\ \texttt{(B)}\ \texttt{S}_b]\!] \tag{6}$$

$$\begin{aligned}
\mathcal{F}^*_{\mathbb{S}}[\![\texttt{while}\ \ell\ \texttt{(B)}\ \texttt{S}_b]\!]\, X \quad \triangleq \quad & \{\langle \ell,\ \rho\rangle \mid \rho \in \mathbb{E}\mathbb{v}\} & \text{(a)}\\
& \cup \{\pi_2\langle \ell',\ \rho\rangle\langle \text{after}[\![\texttt{S}]\!],\ \rho\rangle \mid \pi_2\langle \ell',\ \rho\rangle \in X \wedge \mathcal{B}[\![\texttt{B}]\!]\,\rho = \text{ff} \wedge \ell' = \ell\} & \text{(b)}\\
& \cup \{\pi_2\langle \ell',\ \rho\rangle\langle \text{at}[\![\texttt{S}_b]\!],\ \rho\rangle \cdot \pi_3 \mid \pi_2\langle \ell',\ \rho\rangle \in X \wedge \mathcal{B}[\![\texttt{B}]\!]\,\rho = \text{tt} \wedge & \text{(c)}\\
& \qquad \langle \text{at}[\![\texttt{S}_b]\!],\ \rho\rangle \cdot \pi_3 \in \widehat{\mathcal{S}}^*_{\mathbb{S}}[\![\texttt{S}_b]\!] \wedge \ell' = \ell\}
\end{aligned}$$

(a) either the execution observation stop $\text{at}[\![\texttt{while}\ \ell\ \texttt{(B)}\ \texttt{S}_b]\!] = \ell$, or

(b) after a number of iterations, control is back to ℓ, the test is false, and the loop is exited, or

(c) after a number of iterations, control is back to ℓ, the test is true, and the loop body is executed

(This includes the termination of the loop body $\text{after}[\![\texttt{S}_b]\!] = \text{at}[\![\texttt{while}\ \ell\ \texttt{(B)}\ \texttt{S}_b]\!] = \ell$)

Maximal trace semantics

- Maximal trace semantics

$$\mathcal{S}^{+}[\![\mathsf{S}]\!] \triangleq \{\pi\langle\ell,\ \rho\rangle \in \mathcal{S}^{*}[\![\mathsf{S}]\!] \mid (\ell = \mathsf{after}[\![\mathsf{S}]\!]) \vee (\mathsf{escape}[\![\mathsf{S}]\!] \wedge \ell = \mathsf{break\text{-}to}[\![\mathsf{S}]\!])\}$$
$$\mathcal{S}^{\infty}[\![\mathsf{S}]\!] \triangleq \lim(\mathcal{S}^{*}[\![\mathsf{S}]\!])$$

- Limit

$$\lim \mathcal{T} \triangleq \{\langle\pi,\ \pi'\rangle \mid \pi' \in \mathbb{T}^{\infty} \wedge \forall n \in \mathbb{N}\ .\ \langle\pi,\ \pi'[0..n]\rangle \in \mathcal{T}\}.$$

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 x is always positive or zero during program execution.
- $(? : \texttt{x >= }\underline{\texttt{x}})^*$ states that the value of 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 x should be positive or zero, and next
  - if program point ℓ is ever reached then x should be 0, and next
  - if computations go on after program point ℓ then 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

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
B	$\in$	$\mathbb{B}$	boolean expressions such that $\mathbb{vars}[\![\mathsf{B}]\!] \subseteq \mathbb{V} \cup \underline{\mathbb{V}}$

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

regular expressions (7)

- 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)

$\mathbb{R}^{+\!\!\!|}$ alternative $\mid$-free regular expressions

Semantics of regular expressions

- The semantics $\mathcal{S}^{\mathsf{r}}[\![\mathtt{R}]\!]$ of a regular expression $\mathtt{R}$ 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 $\mathtt{R}$
- Example:
  - $\mathtt{R} \triangleq \ell : \mathtt{x} = \underline{\mathtt{x}} \bullet \ell' : \mathtt{x} = \underline{\mathtt{x}} + 1$
  - $\mathcal{S}^{\mathsf{r}}[\![\mathtt{R}]\!] = \{\langle\underline{\varrho}, \langle\ell_1, \rho\rangle\langle\ell_2, \rho'\rangle\rangle \mid \rho(\mathtt{x}) = \underline{\varrho}(\mathtt{x}) \wedge \rho'(\mathtt{x}) = \underline{\varrho}(\mathtt{x}) + 1\}[]$

  - 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

$$
\begin{aligned}
\mathscr{A}[\![1]\!]\underline{\varrho}, \rho &\triangleq 1 \\
\mathscr{A}[\![\underline{\mathsf{x}}]\!]\underline{\varrho}, \rho &\triangleq \underline{\varrho}(\mathsf{x}) \\
\mathscr{A}[\![\mathsf{x}]\!]\underline{\varrho}, \rho &\triangleq \rho(\mathsf{x}) \\
\mathscr{A}[\![\mathsf{A}_1 - \mathsf{A}_2]\!]\underline{\varrho}, \rho &\triangleq \mathscr{A}[\![\mathsf{A}_1]\!]\underline{\varrho}, \rho - \mathscr{A}[\![\mathsf{A}_2]\!]\underline{\varrho}, \rho \\
\mathscr{B}[\![\mathsf{A}_1 < \mathsf{A}_2]\!]\underline{\varrho}, \rho &\triangleq \mathscr{A}[\![\mathsf{A}_1]\!]\underline{\varrho}, \rho < \mathscr{A}[\![\mathsf{A}_2]\!]\underline{\varrho}, \rho \\
\mathscr{B}[\![\mathsf{B}_1 \text{ nand } \mathsf{B}_2]\!]\underline{\varrho}, \rho &\triangleq \mathscr{B}[\![\mathsf{B}_1]\!]\underline{\varrho}, \rho \uparrow \mathscr{B}[\![\mathsf{B}_2]\!]\underline{\varrho}, \rho
\end{aligned} \tag{8}
$$

Semantics of regular expressions (Cont'd)

Semantics of regular expressions

$$
\begin{aligned}
\mathcal{S}^{\mathsf{r}}[\![\varepsilon]\!] &\triangleq \{\langle \underline{\varrho},\, \ni\rangle \mid \underline{\varrho} \in \mathbb{E}\mathbb{v}\} \\
\mathcal{S}^{\mathsf{r}}[\![\mathsf{L} : \mathsf{B}]\!] &\triangleq \{\langle \underline{\varrho},\, \langle \ell,\, \rho\rangle\rangle \mid \ell \in \mathsf{L} \wedge \mathcal{B}[\![\mathsf{B}]\!]\underline{\varrho}, \rho\} \\
\mathcal{S}^{\mathsf{r}}[\![\mathsf{R}_1\mathsf{R}_2]\!] &\triangleq \mathcal{S}^{\mathsf{r}}[\![\mathsf{R}_1]\!] \odot \mathcal{S}^{\mathsf{r}}[\![\mathsf{R}_2]\!] \\
\mathcal{S} \odot \mathcal{S}' &\triangleq \{\langle \underline{\varrho},\, \pi \cdot \pi'\rangle \mid \langle \underline{\varrho},\, \pi\rangle \in \mathcal{S} \wedge \langle \underline{\varrho},\, \pi'\rangle \in \mathcal{S}'\} \\
\mathcal{S}^{\mathsf{r}}[\![\mathsf{R}_1 \mid \mathsf{R}_2]\!] &\triangleq \mathcal{S}^{\mathsf{r}}[\![\mathsf{R}_1]\!] \cup \mathcal{S}^{\mathsf{r}}[\![\mathsf{R}_2]\!]
\end{aligned}
$$

$$
\begin{aligned}
\mathcal{S}^{\mathsf{r}}[\![\mathsf{R}]\!]^1 &\triangleq \mathcal{S}^{\mathsf{r}}[\![\mathsf{R}]\!] \\
\mathcal{S}^{\mathsf{r}}[\![\mathsf{R}]\!]^{n+1} &\triangleq \mathcal{S}^{\mathsf{r}}[\![\mathsf{R}]\!]^n \odot \mathcal{S}^{\mathsf{r}}[\![\mathsf{R}]\!] \\
\mathcal{S}^{\mathsf{r}}[\![\mathsf{R}^*]\!] &\triangleq \bigcup_{n\in\mathbb{N}} \mathcal{S}^{\mathsf{r}}[\![\mathsf{R}]\!]^n \\
\mathcal{S}^{\mathsf{r}}[\![\mathsf{R}^+]\!] &\triangleq \bigcup_{n\in\mathbb{N}\setminus\{0\}} \mathcal{S}^{\mathsf{r}}[\![\mathsf{R}]\!]^n \\
\mathcal{S}^{\mathsf{r}}[\![(\mathsf{R})]\!] &\triangleq \mathcal{S}^{\mathsf{r}}[\![\mathsf{R}]\!]
\end{aligned}
\qquad (9)
$$

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)

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

where

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

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

Abstraction

Model checking is an boolean abstraction of the program semantics

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

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

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

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

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

- 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 $\mathrm{dnf}(\mathtt{R}) \doteqdot \mathtt{R}$)

$$\mathrm{dnf}(\varepsilon) \triangleq \varepsilon \qquad \mathrm{dnf}(\mathtt{L} : \mathtt{B}) \triangleq \mathtt{L} : \mathtt{B}$$

$$\mathrm{dnf}(\mathtt{R}_1 \mid \mathtt{R}_2) \triangleq \mathrm{dnf}(\mathtt{R}_1) \mid \mathrm{dnf}(\mathtt{R}_2) \qquad \mathrm{dnf}(\mathtt{R}^+) \triangleq \mathrm{dnf}(\mathtt{R}\mathtt{R}^*)$$

$$\mathrm{dnf}(\mathtt{R}^*) \triangleq \text{let } \mathtt{R}^1 \mid \ldots \mid \mathtt{R}^n = \mathrm{dnf}(\mathtt{R}) \text{ in } ((\mathtt{R}^1)^* \ldots (\mathtt{R}^n)^*)^* \qquad \mathrm{dnf}((\mathtt{R})) \triangleq (\mathrm{dnf}(\mathtt{R}))$$

$$\mathrm{dnf}(\mathtt{R}_1\mathtt{R}_2) \triangleq \text{let } \mathtt{R}_1^1 \mid \ldots \mid \mathtt{R}_1^{n_1} = \mathrm{dnf}(\mathtt{R}_1) \text{ and } \mathtt{R}_2^1 \mid \ldots \mid \mathtt{R}_2^{n_2} = \mathrm{dnf}(\mathtt{R}_2) \text{ in } \big|_{i=1}^{n_1} \big|_{j=1}^{n_2} \mathtt{R}_1^i \mathtt{R}_2^j$$

[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 $\mathtt{R} \in \mathbb{R}^{+} \cap \mathbb{R}^{\dagger}$, 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 $\mathtt{R}$.

$$
\begin{array}{rcll}
\mathsf{fstnxt}(\mathtt{L} : \mathtt{B}) & \triangleq & \langle \mathtt{L} : \mathtt{B},\ \varepsilon \rangle & (10)\\
\mathsf{fstnxt}(\mathtt{R}_1\mathtt{R}_2) & \triangleq & \mathsf{fstnxt}(\mathtt{R}_2) & \text{if } \mathtt{R}_1 \in \mathbb{R}_\varepsilon\\
\mathsf{fstnxt}(\mathtt{R}_1\mathtt{R}_2) & \triangleq & \text{let } \langle \mathtt{R}_1^f,\ \mathtt{R}_1^n \rangle = \mathsf{fstnxt}(\mathtt{R}_1) \text{ in } [\![\mathtt{R}_1^n \in \mathbb{R}_\varepsilon \mathrel{?} \langle \mathtt{R}_1^f,\ \mathtt{R}_2 \rangle \mathrel{\text{⨾}} \langle \mathtt{R}_1^f,\ \mathtt{R}_1^n \bullet \mathtt{R}_2 \rangle]\!] & \\
 & & & \text{if } \mathtt{R}_1 \notin \mathbb{R}_\varepsilon\\
\mathsf{fstnxt}(\mathtt{R}^{+}) & \triangleq & \text{let } \langle \mathtt{R}^f,\ \mathtt{R}^n \rangle = \mathsf{fstnxt}(\mathtt{R}) \text{ in } [\![\mathtt{R}^n \in \mathbb{R}_\varepsilon \mathrel{?} \langle \mathtt{R}^f,\ \mathtt{R}^{*} \rangle \mathrel{\text{⨾}} \langle \mathtt{R}^f,\ \mathtt{R}^n \bullet \mathtt{R}^{*} \rangle]\!] & \\
\mathsf{fstnxt}((\mathtt{R})) & \triangleq & \mathsf{fstnxt}(\mathtt{R}) &
\end{array}
$$

Calculational design of the abstract interpreter (I)

Methodology

- Apply the abstraction function

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

  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 2 of regular model checking

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

$$\mathcal{M}^{t}\langle\underline{\varrho},\ \varepsilon\rangle\pi \triangleq \langle\mathrm{tt},\ \varepsilon\rangle \tag{11}$$

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

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

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

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

Definition 2 of regular model checking (Cont'd)

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

$$\mathcal{M}^+\langle\underline{\varrho},\ \mathsf{R}\rangle\Pi \triangleq \{\langle\pi,\ \mathsf{R}'\rangle \mid \pi \in \Pi \wedge \langle \mathsf{tt},\ \mathsf{R}'\rangle = \mathcal{M}^t\langle\underline{\varrho},\ \mathsf{R}\rangle\pi\} \qquad (12)$$

  This abstraction is a Galois connection

$$\langle\wp(\mathbb{S}^+),\ \subseteq\rangle \xleftarrow[\mathcal{M}^+\langle\underline{\varrho},\ \mathsf{R}\rangle]{\gamma_{\mathcal{M}^+\langle\underline{\varrho},\ \mathsf{R}\rangle}} \langle\wp(\mathbb{S}^+ \times \mathbb{R}^+),\ \subseteq\rangle \quad \text{for } \mathsf{R} \in \mathbb{R}^+ \text{ in (12)} \qquad (16)$$

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

$$\mathcal{M}^+[\![\mathsf{S}]\!]\langle\underline{\varrho},\ \mathsf{R}\rangle \triangleq \mathcal{M}^+\langle\underline{\varrho},\ \mathsf{R}\rangle(\widehat{\mathcal{S}}^{\,*}_{\mathbb{S}}[\![\mathsf{S}]\!]) \qquad (13)$$

Definition 2 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 $\mathsf{R} \in \mathbb{R}$):

$$\mathcal{M}\langle\underline{\varrho},\ \mathsf{R}\rangle\Pi \triangleq \text{let}\,(\mathsf{R}_1 \mid \ldots \mid \mathsf{R}_n) = \text{dnf}(\mathsf{R}) \text{ in} \quad \bigcup_{i=1}^{n} \{\pi \mid \exists \mathsf{R}' \in \mathbb{R} \,.\, \langle\pi,\ \mathsf{R}'\rangle \in \mathcal{M}^{\dagger}\langle\underline{\varrho},\ \mathsf{R}_i\rangle\Pi\} \tag{14}$$

This abstraction is a Galois connection

$$\langle\wp(\mathbb{S}^+),\ \subseteq\rangle \xrightleftharpoons[\mathcal{M}\langle\underline{\varrho},\ \mathsf{R}\rangle]{\gamma_{\mathcal{M}\langle\underline{\varrho},\ \mathsf{R}\rangle}} \langle\wp(\mathbb{S}^+),\ \subseteq\rangle \qquad \text{for } \mathsf{R} \in \mathbb{R} \text{ in (14)} \tag{17}$$

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

$$\mathcal{M}[\![\mathsf{S}]\!]\langle\underline{\varrho},\ \mathsf{R}\rangle \triangleq \mathcal{M}\langle\underline{\varrho},\ \mathsf{R}\rangle(\widehat{\mathcal{S}}^{\,*}_{\mathbb{S}}[\![\mathsf{S}]\!]) \tag{15}$$

Definition 2 of regular model checking (Cont'd)

- Back to boolean model-checking

$$\langle \wp(\mathbb{S}^+),\ \subseteq\rangle \xrightleftharpoons[\alpha_{\mathcal{M}\langle\underline{\varrho},\ \mathsf{R}\rangle}]{\gamma_{\mathcal{M}\langle\underline{\varrho},\ \mathsf{R}\rangle}} \langle \mathbb{B},\ \Leftarrow\rangle \tag{18}$$

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 $\mathscr{M}^t\langle\underline{\varrho}, \mathsf{R}\rangle(\pi\mathbin{\frown}\pi') = \langle\mathsf{tt}, \mathsf{R}'\rangle \Leftrightarrow (\exists \mathsf{R}'' \in \mathbb{R} \,.\, \mathscr{M}^t\langle\underline{\varrho}, \mathsf{R}\rangle(\pi) = \langle\mathsf{tt}, \mathsf{R}''\rangle \wedge \mathscr{M}^t\langle\underline{\varrho}, \mathsf{R}''\rangle(\pi') = \langle\mathsf{tt}, \mathsf{R}'\rangle)$. □

- Structural model checking

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

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

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

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

Calculational design

$$
\begin{aligned}
& \mathcal{M}[\![\mathsf{S}]\!]\langle \underline{\varrho},\ \mathsf{R}\rangle \\
\triangleq\ & \mathcal{M}\langle \underline{\varrho},\ \mathsf{R}\rangle(\widehat{\mathcal{S}}^{\,*}_{\mathbb{S}}[\![\mathsf{S}]\!]) && \text{\{(15)\}} \\
=\ & \mathcal{M}\langle \underline{\varrho},\ \mathsf{R}\rangle(\widehat{\mathcal{T}}_{\mathcal{S}}[\![\mathsf{S}]\!](\prod_{\mathsf{S}' \triangleleft \mathsf{S}} \widehat{\mathcal{S}}^{\,*}_{\mathbb{S}}[\![\mathsf{S}']\!])\langle \underline{\varrho},\ \mathsf{R}\rangle)
\end{aligned}
$$

{by structural definition $\widehat{\mathcal{S}}^{\,*}_{\mathbb{S}}[\![\mathsf{S}]\!] = \widehat{\mathcal{T}}_{\mathcal{S}}[\![\mathsf{S}]\!](\prod_{\mathsf{S}' \triangleleft \mathsf{S}} \widehat{\mathcal{S}}^{\,*}_{\mathbb{S}}[\![\mathsf{S}']\!])$ of the stateful prefix trace semantics in Section **2**}

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

$$
= \widehat{\mathcal{T}}_{\mathcal{M}}[\![\mathsf{S}]\!](\prod_{\mathsf{S}' \triangleleft \mathsf{S}} \mathcal{M}[\![\mathsf{S}']\!])\langle \underline{\varrho},\ \mathsf{R}\rangle
$$

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

$$
\begin{aligned}
=\ & \widehat{\mathcal{T}}_{\mathcal{M}}[\mathsf{S}](\prod_{\mathsf{S}' \triangleleft \mathsf{S}} \widehat{\mathcal{M}}[\![\mathsf{S}']\!])\langle \underline{\varrho},\ \mathsf{R}\rangle && \text{\{ind. hyp.\}} \\
\triangleq\ & \widehat{\mathcal{M}}[\![\mathsf{S}]\!]\langle \underline{\varrho},\ \mathsf{R}\rangle && \text{\{by defining } \widehat{\mathcal{M}}[\![\mathsf{S}]\!] \triangleq \widehat{\mathcal{T}}_{\mathcal{M}}[\![\mathsf{S}]\!](\textstyle\prod_{\mathsf{S}' \triangleleft \mathsf{S}} \widehat{\mathcal{M}}[\![\mathsf{S}']\!])\text{\}}
\end{aligned}
$$

Calculational design

For iteration statements, $\widehat{\mathcal{T}}[\![\mathsf{S}]\!](\prod_{\mathsf{S}' \triangleleft \mathsf{S}} \widehat{\mathcal{S}}^{\,*}_{\mathbb{S}}[\![\mathsf{S}']\!])\langle \underline{\varrho}, \mathsf{R}\rangle$ 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 $\langle C, \sqsubseteq, \bot, \top, \sqcup, \sqcap\rangle$ and $\langle \mathcal{A}, \preccurlyeq, 0, 1, \curlyvee, \curlywedge\rangle$ are complete lattices, $f \in C \xrightarrow{\nearrow} C$ is increasing, $\langle C, \sqsubseteq\rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle \mathcal{A}, \preccurlyeq\rangle$, $\overline{f} \in \mathcal{A} \xrightarrow{\nearrow} \mathcal{A}$ 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

$$
\begin{array}{lll}
& \mathcal{M}^{+}[\![\mathsf{s}]\!]\langle \underline{\varrho},\ \varepsilon\rangle & \\
\triangleq & \mathcal{M}^{+}\langle \underline{\varrho},\ \varepsilon\rangle(\widehat{\mathcal{S}}^{*}_{\mathbb{S}}[\![\mathsf{s}]\!]) & \text{(13)} \\
\triangleq & \{\langle \pi,\ \varepsilon'\rangle \mid \pi \in \widehat{\mathcal{S}}^{*}_{\mathbb{S}}[\![\mathsf{s}]\!] \wedge \langle \mathrm{tt},\ \varepsilon'\rangle = \mathcal{M}^{t}\langle \underline{\varrho},\ \varepsilon\rangle\pi\} & \text{(12)} \\
\triangleq & \{\langle \pi,\ \varepsilon'\rangle \mid \pi \in \widehat{\mathcal{S}}^{*}_{\mathbb{S}}[\![\mathsf{s}]\!] \wedge \langle \mathrm{tt},\ \varepsilon'\rangle = \langle \mathrm{tt},\ \varepsilon\rangle\} & \mathcal{M}^{t}\langle \underline{\varrho},\ \varepsilon\rangle\pi \triangleq \langle \mathrm{tt},\ \varepsilon\rangle \text{ by (11)} \\
= & \{\langle \pi,\ \varepsilon\rangle \mid \pi \in \widehat{\mathcal{S}}^{*}_{\mathbb{S}}[\![\mathsf{s}]\!]\} & \text{def. } = \\
\triangleq & \widehat{\mathcal{M}}^{+}[\![\mathsf{s}]\!]\langle \underline{\varrho},\ \varepsilon\rangle &
\end{array}
$$

Definition 3 (Structural model checking)

- Model checking an empty temporal specification ε.

$$\widehat{\mathcal{M}}^{+}[\![\mathsf{s}]\!]\langle \underline{\varrho},\ \varepsilon\rangle \triangleq \{\langle \pi,\ \varepsilon\rangle \mid \pi \in \widehat{\mathcal{S}}^{*}_{\mathbb{S}}[\![\mathsf{s}]\!]\} \tag{20}$$

Structural regular model checking of programs P ::= Sl

$$
\begin{array}{lll}
& \mathcal{M}[\![\mathtt{P}]\!]\langle \underline{\varrho},\ \mathtt{R}\rangle & \\
\triangleq & \mathcal{M}\langle \underline{\varrho},\ \mathtt{R}\rangle(\widehat{\mathcal{S}}^{\,*}_{\mathbb{S}}[\![\mathtt{P}]\!]) & \wr(15)\wr \\
\triangleq & \text{let}\,(\mathtt{R}_1 \mid \ldots \mid \mathtt{R}_n) = \text{dnf}(\mathtt{R}) \text{ in } \bigcup_{i=1}^{n} \{\pi \mid \exists \mathtt{R}' \in \mathbb{R} \,.\, \langle \pi,\ \mathtt{R}'\rangle \in \mathcal{M}^{+}\langle \underline{\varrho},\ \mathtt{R}_i\rangle(\widehat{\mathcal{S}}^{\,*}_{\mathbb{S}}[\![\mathtt{P}]\!])\} & \wr(14)\wr \\
= & \text{let}\,(\mathtt{R}_1 \mid \ldots \mid \mathtt{R}_n) = \text{dnf}(\mathtt{R}) \text{ in } \bigcup_{i=1}^{n} \{\pi \mid \exists \mathtt{R}' \in \mathbb{R} \,.\, \langle \pi,\ \mathtt{R}'\rangle \in \mathcal{M}^{+}\langle \underline{\varrho},\ \mathtt{R}_i\rangle(\widehat{\mathcal{S}}^{\,*}_{\mathbb{S}}[\![\mathtt{Sl}]\!])\} & \\
& & \wr\widehat{\mathcal{S}}^{\,*}_{\mathbb{S}}[\![\mathtt{P}]\!] = \widehat{\mathcal{S}}^{\,*}_{\mathbb{S}}[\![\mathtt{Sl}]\!]\wr \\
= & \text{let}\,(\mathtt{R}_1 \mid \ldots \mid \mathtt{R}_n) = \text{dnf}(\mathtt{R}) \text{ in } \bigcup_{i=1}^{n} \{\pi \mid \exists \mathtt{R}' \in \mathbb{R} \,.\, \langle \pi,\ \mathtt{R}'\rangle \in \widehat{\mathcal{M}}^{+}\langle \underline{\varrho},\ \mathtt{R}_i\rangle(\widehat{\mathcal{S}}^{\,*}_{\mathbb{S}}[\![\mathtt{Sl}]\!])\} & \\
& & \wr\text{ind. hyp.}\wr \\
= & \text{let}\,(\mathtt{R}_1 \mid \ldots \mid \mathtt{R}_n) = \text{dnf}(\mathtt{R}) \text{ in } \bigcup_{i=1}^{n} \{\pi \mid \exists \mathtt{R}' \in \mathbb{R} \,.\, \langle \pi,\ \mathtt{R}'\rangle \in \widehat{\mathcal{M}}^{+}[\![\mathtt{Sl}]\!]\langle \underline{\varrho},\ \mathtt{R}_i\rangle\} & \wr(13)\wr \\
\triangleq & \widehat{\mathcal{M}}[\![\mathtt{Sl}]\!]\langle \underline{\varrho},\ \mathtt{R}\rangle & \wr(19)\wr
\end{array}
$$

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

Definition 3 (Structural model checking, contn'd)

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

$$\widehat{\mathcal{M}}[\![\texttt{P}]\!]\langle \underline{\varrho}, \texttt{R}\rangle \triangleq \text{let}\,(\texttt{R}_1 \mid \ldots \mid \texttt{R}_n) = \text{dnf}(\texttt{R}) \text{ in} \bigcup_{i=1}^{n} \{\pi \mid \exists \texttt{R}' \in \mathbb{R} \,.\, \langle \pi, \texttt{R}'\rangle \in \widehat{\mathcal{M}}^{+}[\![\texttt{Sl}]\!]\langle \underline{\varrho}, \texttt{R}_i\rangle\} \tag{19}$$

Structural regular model checking of assignments S ::= ℓ x = A ;

Definition 3 (Structural model checking, contn'd)

- Model checking an assignment statement S ::= ℓ x = A ;

$$
\begin{aligned}
\widehat{\mathscr{M}}^{+}[\![\mathsf{S}]\!]\langle \underline{\varrho}, \mathsf{R}\rangle \triangleq\ & \text{let } \langle \mathsf{L} : \mathsf{B}, \mathsf{R}'\rangle = \mathsf{fstnxt}(\mathsf{R}) \text{ in} & (23)\\
& \{\langle\langle \mathsf{at}[\![\mathsf{S}]\!], \rho\rangle, \mathsf{R}'\rangle \mid \langle \underline{\varrho}, \langle \mathsf{at}[\![\mathsf{S}]\!], \rho\rangle\rangle \in \mathcal{S}^{r}[\![\mathsf{L} : \mathsf{B}]\!]\} & \text{(a)}\\
& \cup \{\langle\langle \mathsf{at}[\![\mathsf{S}]\!], \rho\rangle\langle \mathsf{after}[\![\mathsf{S}]\!], \rho[\mathsf{x} \leftarrow \mathscr{A}[\![\mathsf{A}]\!]\rho]\rangle, \varepsilon\rangle \mid \mathsf{R}' \in \mathbb{R}_{\varepsilon} \wedge & \text{(b)}\\
& \qquad \langle \underline{\varrho}, \langle \mathsf{at}[\![\mathsf{S}]\!], \rho\rangle\rangle \in \mathcal{S}^{r}[\![\mathsf{L} : \mathsf{B}]\!]\}\\
& \cup \{\langle\langle \mathsf{at}[\![\mathsf{S}]\!], \rho\rangle\langle \mathsf{after}[\![\mathsf{S}]\!], \rho[\mathsf{x} \leftarrow \mathscr{A}[\![\mathsf{A}]\!]\rho]\rangle, \mathsf{R}''\rangle \mid \mathsf{R}' \notin \mathbb{R}_{\varepsilon} \wedge & \text{(c)}\\
& \qquad \langle \underline{\varrho}, \langle \mathsf{at}[\![\mathsf{S}]\!], \rho\rangle\rangle \in \mathcal{S}^{r}[\![\mathsf{L} : \mathsf{B}]\!] \wedge \langle \mathsf{L}' : \mathsf{B}', \mathsf{R}''\rangle = \mathsf{fstnxt}(\mathsf{R}') \wedge\\
& \qquad \langle \underline{\varrho}, \langle \mathsf{after}[\![\mathsf{S}]\!], \rho[\mathsf{x} \leftarrow \mathscr{A}[\![\mathsf{A}]\!]\rho]\rangle\rangle \in \mathcal{S}^{r}[\![\mathsf{L}' : \mathsf{B}']\!]\}
\end{aligned}
$$

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

$$
\begin{aligned}
& \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\} && \text{(13) and (12)} \\
= {} & \{\langle \pi,\ \mathtt{R}'\rangle \mid \pi \in \{\langle \ell,\ \rho\rangle \mid \rho \in \mathbb{E}\mathbb{v}\} \cup \{\langle \ell,\ \rho\rangle\langle \mathrm{after}[\![\mathtt{S}]\!],\ \rho[\mathtt{x} \leftarrow v]\rangle \mid \rho \in \mathbb{E}\mathbb{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\} && \text{(1)} \\
= {} & \{\langle\langle \ell,\ \rho\rangle,\ \mathtt{R}'\rangle \mid \rho \in \mathbb{E}\mathbb{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}\mathbb{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\} && \text{def. } \cup \text{ and } \in \\
= {} & \{\langle\langle \ell,\ \rho\rangle,\ \mathtt{R}'\rangle \mid \langle \mathrm{tt},\ \mathtt{R}'\rangle = \mathrm{let}\ \langle \mathtt{L} : \mathtt{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} : \mathtt{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} : \mathtt{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} : \mathtt{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\, |\!)\} && \text{(11)} \\
= {} & \ldots
\end{aligned}
$$

$$
\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{⸗}\ \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\,|\!)\} \\
&\hfill \text{(def. = and } \mathcal{M}^{t}\langle \underline{\varrho},\ \varepsilon\rangle\pi \triangleq \langle \mathsf{tt},\ \varepsilon\rangle \text{ by (11))} \\
&= \{\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{⸗}\ \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}']\!]\,|\!)\} \hfill \text{((11))} \\
&= \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}']\!]\} \hfill \text{(def. } \cup\text{)} \\
&= \widehat{\mathcal{M}}^{+}[\![\mathsf{S}]\!]\ \langle \underline{\varrho},\ \mathsf{R}\rangle \hfill \text{((23))} \quad \square
\end{aligned}
$$

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

Definition 3 (Structural model checking, contn'd)

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

$$
\begin{aligned}
\widehat{\mathcal{M}}^{+}[\![\texttt{Sl}]\!]\langle\underline{\varrho},\ \texttt{R}\rangle \quad \triangleq \quad & \widehat{\mathcal{M}}^{+}[\![\texttt{Sl}']\!]\langle\underline{\varrho},\ \texttt{R}\rangle \\
& \cup \{\langle\pi\cdot\langle\texttt{at}[\![\texttt{S}]\!],\ \rho\rangle\cdot\pi',\ \texttt{R}''\rangle \mid \langle\pi\cdot\langle\texttt{at}[\![\texttt{S}]\!],\ \rho\rangle,\ \texttt{R}'\rangle \in \widehat{\mathcal{M}}^{+}[\![\texttt{Sl}']\!]\langle\underline{\varrho},\ \texttt{R}\rangle \wedge \\
& \qquad \langle\langle\texttt{at}[\![\texttt{S}]\!],\ \rho\rangle\cdot\pi',\ \texttt{R}''\rangle \in \widehat{\mathcal{M}}^{+}[\![\texttt{S}]\!]\langle\underline{\varrho},\ \texttt{R}'\rangle\}
\end{aligned}
\tag{21}
$$

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

Definition 3 (Structural model checking, contn'd)

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

$$\widehat{\mathcal{M}}^{\dagger}[\![\mathtt{S}]\!]\langle \underline{\varrho},\ \mathtt{R}\rangle \quad \triangleq \quad \mathsf{lfp}^{\subseteq}\,(\widehat{\mathcal{F}}^{\dagger}[\![\mathtt{S}]\!]\langle \underline{\varrho},\ \mathtt{R}\rangle) \tag{26}$$

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

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 (32).

Cousot, Patrick and Radhia Cousot (1977). “Abstract Interpretation: A Unified Lattice Model for Static Analysis of Programs by Construction or Approximation of Fixpoints”. In: *POPL*. ACM, pp. 238–252 (3, 4).

– (1979). “Systematic Design of Program Analysis Frameworks”. In: *POPL*. ACM Press, pp. 269–282 (3, 4, 45).

– (2000). “Temporal Abstract Interpretation”. In: *POPL*. ACM, pp. 12–25 (59).

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 (59).

Schneider, Fred B. (2000). “Enforceable security policies”. *ACM Trans. Inf. Syst. Secur.* 3.1, pp. 30–50 (19).

References II

Wolper, Pierre (1983). "Temporal Logic Can Be More Expressive". *Information and Control* 56.1/2, pp. 72–99 (3, 4).

The End, Thank you