Calculational Design of a Regular Model Checker by Abstract Interpretation*

Patrick Cousot

CS, CIMS, NYU, New York, NY, USA, visiting IMDEA Software, Madrid, Spain

Abstract

Security monitors have been used to check for safety program properties at runtime, that is for any given execution trace. Such security monitors check a safety temporal property specified by a finite automaton or, equivalently, a regular expression. Checking this safety temporal specification for all possible execution traces, that is the program semantics, is a static analysis problem, more precisely a model checking problem, since model checking specializes in temporal properties. We show that the model checker can be formally designed by calculus, by abstract interpretation of a formal trace semantics of the programming language. The result is a structural sound and complete model checker, which proceeds by induction on the program syntax (as opposed to the more classical approach using computation steps formalized by a transition system). By Rice theorem, further hypotheses or abstractions are needed to get a realistic model checking algorithm.

Keywords: Abstract interpretation, Calculational design, Model checking.

1. Introduction

Model checking [9, 43] consists in proving that a model of a given program/computer system satisfies a temporal specification[1]. Traditionally, the model of the given program/computer system is a transition system and its semantics is the set of traces generated by the transition system. The temporal specification is usually one of the many variants of temporal logics such as the Linear Time Temporal logic (LTL) [42], the Computation Tree Logic (CTL)[9], or the combination CTL* of the two [21]. The semantics of the temporal specification is a set of traces. The problem is therefore to check that the set of traces of the semantics of the given program/computer system is included in the set of traces

*Supported by NSF Grant CCF-1617717.

Email address: pcousot@cims.nyu.edu (Patrick Cousot)

[1] We define model checking as the verification of temporal properties and do not reduce it to the reachability analysis (as done *e.g.* in [10, Ch. 15, 16, 17, *etc.*]) since reachability analysis predates model checking [14] including for the use of transition systems [12].

of the semantics of the temporal specification. This is a Galois connection-based abstraction and so a model checking algorithm is an abstract interpretation of the program semantics that can be designed by calculus. To show that we consider a non-conventional temporal specification using regular expressions [33] and a structural fixpoint prefix-closed trace semantics which differs from the traditional small-step operational semantics specified by a transition system. There are properties of traces that are not expressible in temporal logic but are easily expressible using regular expressions [49].

2. Syntax and Trace Semantics of the Programming Language

2.1. Syntax

Programs are a subset of `C` with the following context-free syntax.

$\mathtt{x}, \mathtt{y}, \ldots$	$\in$	$\mathbb{X}$	variable ($\mathbb{X}$ not empty)
$\mathtt{A}$	$\in$	$\mathbb{A}$ ::= `1` | `x` | $\mathtt{A}_1$ `-` $\mathtt{A}_2$	arithmetic expression[2]
$\mathtt{B}$	$\in$	$\mathbb{B}$ ::= $\mathtt{A}_1$ `<` $\mathtt{A}_2$ | $\mathtt{B}_1$ `nand` $\mathtt{B}_2$	boolean expression
$\mathtt{E}$	$\in$	$\mathbb{E}$ ::= `A` | `B`	expression
$\mathtt{S}$	$\in$	$\mathbb{S}$ ::=	statement
		`x = A ;`	assignment
		| `;`	skip
		| `if (B) S` | `if (B) S else S`	conditionals
		| `while (B) S` | `break ;`	iteration and break
		| `{ Sl }`	compound statement
$\mathtt{Sl} \in \mathbb{SI}$		::= `Sl S` | ϵ	statement list
$\mathtt{P} \in \mathbb{P}$		::= `Sl`	program

A break exits the closest enclosing loop, if none this is a syntactic error. If `P` is a program then `int main () { P }` is a valid `C` program after adding variable declarations. We call "[program] component" $\mathtt{S} \in \mathbb{P}\mathtt{c} \triangleq \mathbb{S} \cup \mathbb{SI} \cup \mathbb{P}$ either a statement, a statement list, or a program. We let $\lhd$ be the syntactic relation between immediate syntactic components, as defined in **Appendix A**. For example, if $\mathtt{S}$ ::= `if (B)` $\mathtt{S}_t$ `else` $\mathtt{S}_f$ then $\mathtt{B} \lhd \mathtt{S}$, $\mathtt{S}_t \lhd \mathtt{S}$, and $\mathtt{S}_f \lhd \mathtt{S}$ while for $\mathtt{Sl}$ ::= $\mathtt{Sl}'$ `S` \| ϵ, we have $\mathtt{Sl}' \lhd \mathtt{Sl}$, $\mathtt{S} \lhd \mathtt{Sl}$, and $\epsilon \lhd \mathtt{Sl}$.

2.2. Program labels

Labels $\ell \in \mathbb{L}$ are not part of the language, but useful to discuss program points reached during execution. For each program component `S`, we define

[2] All arithmetic operations can be defined from `1`, `-`, and iteration. Same for boolean expressions with `<` and `nand`.

$\mathsf{at}[\![\mathtt{S}]\!]$	the program point at which execution of `S` starts;
$\mathsf{aft}[\![\mathtt{S}]\!]$	the program exit point after `S`, at which execution of `S` is supposed to normally terminate, if ever;
$\mathsf{esc}[\![\mathtt{S}]\!]$	a boolean indicating whether or not the program component `S` contains a **`break ;`** statement escaping out of that component `S`;
$\mathsf{brk\text{-}to}[\![\mathtt{S}]\!]$	the program point at which execution of the program component `S` goes to when a **`break ;`** statement escapes out of that component `S`;
$\mathsf{brks\text{-}of}[\![\mathtt{S}]\!]$	the set of labels of all **`break ;`** statements that can escape out of `S`;
$\mathsf{in}[\![\mathtt{S}]\!]$	the set of program points inside `S` (including $\mathsf{at}[\![\mathtt{S}]\!]$ but excluding $\mathsf{aft}[\![\mathtt{S}]\!]$ (when `S` is not empty) and $\mathsf{brk\text{-}to}[\![\mathtt{S}]\!]$);
$\mathsf{labs}[\![\mathtt{S}]\!]$	the potentially reachable program points while executing `S` either at, in, or after the statement, or resulting from a break.

Formal definitions are given in **Appendix B**. The program labelling has the following properties.

Lemma 1. For all program components $\mathtt{S} \in \mathscr{PC}$ of a program `P`, $\mathsf{at}[\![\mathtt{S}]\!] \in \mathsf{in}[\![\mathtt{S}]\!]$.

Lemma 2. For all program non-empty components $\mathtt{S} \neq \mathtt{\{}\ \ldots\mathtt{\{}\ \epsilon\ \mathtt{\}}\ldots\ \mathtt{\}}$ of a program `P`, $\mathsf{aft}[\![\mathtt{S}]\!] \notin \mathsf{in}[\![\mathtt{S}]\!]$.

2.3. Prefix trace semantics

Prefix traces are non-empty finite sequences $\pi \in \mathbb{S}^+$ of states where states $\langle \ell, \rho \rangle \in \mathbb{S} \triangleq (\mathbb{L} \times \mathbb{E}\mathtt{v})$ are pairs of a program label $\ell \in \mathbb{S}$ designating the next action to be executed in the program and an environment $\rho \in \mathbb{E}\mathtt{v} \triangleq \mathbb{X} \to \mathbb{V}$ assigning values $\rho(\mathtt{x}) \in \mathbb{V}$ to variables $\mathtt{x} \in \mathbb{X}$. A trace π can be finite $\pi \in \mathbb{S}^+$ or infinite $\pi \in \mathbb{S}^\infty$ (recording a non-terminating computation) so $\mathbb{S}^{+\infty} \triangleq \mathbb{S}^+ \cup \mathbb{S}^\infty$. Trace concatenation $\frown$ is defined as follows

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

The first state $\langle \ell, \rho \rangle$ in a trace $\langle \ell, \rho \rangle \pi$ is called the initial state and $\rho(\mathtt{x})$ the initial value of variable $\mathtt{x} \in \mathbb{X}$. In pattern matching, we sometimes need the empty trace $\ni$. For example if $\sigma\pi\sigma' = \sigma$ then $\pi = \ni$ and so $\sigma = \sigma'$.

2.4. Formal definition of the prefix trace semantics

The prefix trace semantics $\mathscr{S}^*[\![\mathtt{S}]\!]$ is given structurally (by induction on the syntax) using fixpoints for the iteration.

- *The prefix traces of an assignment statement* $\mathtt{S} ::= \ell\ \mathtt{x = A\ ;}$ (where $\mathsf{at}[\![\mathtt{S}]\!] = \ell$) either stops in an initial state in $\{\langle \ell, \rho \rangle \mid \rho \in \mathbb{E}\mathtt{v}\}$ or is this initial state $\langle \ell, \rho \rangle$ followed by the next state $\langle \mathsf{aft}[\![\mathtt{S}]\!], \rho[\mathtt{x} \leftarrow \mathscr{A}[\![\mathtt{A}]\!]\rho] \rangle$ recording the assignment of the value $\mathscr{A}[\![\mathtt{A}]\!]\rho$ of the arithmetic expression to variable `x` when reaching the label $\mathsf{aft}[\![\mathtt{S}]\!]$ after the assignment.

$$\mathcal{S}^*[\![\mathtt{S}]\!] \quad = \quad \{\langle \ell,\ \rho\rangle \mid \rho \in \mathbb{E}\mathtt{v}\} \cup \{\langle \ell,\ \rho\rangle\langle \mathsf{aft}[\![\mathtt{S}]\!],\ \rho[\mathtt{x} \leftarrow \mathcal{A}[\![\mathtt{A}]\!]\rho]\rangle \mid \rho \in \mathbb{E}\mathtt{v}\} \qquad (1)$$

The value of an arithmetic expression $\mathtt{A}$ in environment $\rho \in \mathbb{E}\mathtt{v} \triangleq \mathbb{X} \to \mathbb{V}$ is $\mathcal{A}[\![\mathtt{A}]\!]\rho \in \mathbb{V}$:

$$\mathcal{A}[\![\mathtt{1}]\!]\rho \triangleq 1 \qquad \mathcal{A}[\![\mathtt{x}]\!]\rho \triangleq \rho(\mathtt{x}) \qquad \mathcal{A}[\![\mathtt{A}_1 - \mathtt{A}_2]\!]\rho \triangleq \mathcal{A}[\![\mathtt{A}_1]\!]\rho - \mathcal{A}[\![\mathtt{A}_2]\!]\rho \qquad (2)$$

- *The prefix trace semantics of a break statement* $\mathtt{S} ::= \ell$ **break ;** either stops at ℓ or goes on to the break label $\mathsf{brk\text{-}to}[\![\mathtt{S}]\!]$ (which is defined as the exit label of the closest enclosing iteration).

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

- *The prefix trace semantics of a conditional statement* $\mathtt{S} ::= \mathtt{if}\ \ell\ \mathtt{(B)}\ \mathtt{S}_t$ where $\mathsf{at}[\![\mathtt{S}]\!] = \ell$ is
  - either the trace $\langle \ell,\ \rho\rangle$ when the observation of the execution stops on entry of the program component for initial environment ρ;
  - or, when the value of the boolean expression $\mathtt{B}$ for ρ is false $\mathtt{ff}$, the initial state $\langle \ell,\ \rho\rangle$ followed by the state $\langle \mathsf{aft}[\![\mathtt{S}]\!],\ \rho\rangle$ at the label $\mathsf{aft}[\![\mathtt{S}]\!]$ after the conditional statement;
  - or finally, when the value of the boolean expression $\mathtt{B}$ for ρ is true $\mathtt{tt}$, the initial state$\langle \ell,\ \rho\rangle$ followed by a prefix trace of $\mathtt{S}_t$ starting $\mathsf{at}[\![\mathtt{S}_t]\!]$ in environment ρ (and possibly ending $\mathsf{aft}[\![\mathtt{S}_t]\!] = \mathsf{aft}[\![\mathtt{S}]\!]$).

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

Observe that definition (4) includes the case of a conditional within an iteration and containing a break statement in the true branch $\mathtt{S}_t$. Since $\mathsf{brk\text{-}to}[\![\mathtt{S}]\!] = \mathsf{brk\text{-}to}[\![\mathtt{S}_t]\!]$, from $\langle \mathsf{at}[\![\mathtt{S}_t]\!],\ \rho\rangle\pi\langle \mathsf{brk\text{-}to}[\![\mathtt{S}_t]\!],\ \rho'\rangle \in \mathcal{S}^*[\![\mathtt{S}_t]\!]$ and $\mathcal{B}[\![\mathtt{B}]\!]\rho = \mathtt{tt}$, we infer that $\langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle\langle \mathsf{at}[\![\mathtt{S}_t]\!],\ \rho\rangle\pi\langle \mathsf{brk\text{-}to}[\![\mathtt{S}]\!],\ \rho'\rangle \in \mathcal{S}^*[\![\mathtt{S}]\!]$.

- A prefix finite trace of a conditional statement $\mathtt{if}\ \ell\ \mathtt{(B)}\ \mathtt{S}_t\ \mathtt{else}\ \mathtt{S}_f$ where $\mathsf{at}[\![\mathtt{S}]\!] = \ell$ either stops at ℓ in case (5.a) or, in case (5.b), is the test event $\mathtt{B}$ (respectively $\neg(\mathtt{B})$ in case (5.c)) at ℓ followed by a prefix trace of $\mathtt{S}_t$ (respectively $\mathtt{S}_f$) when boolean expression $\mathtt{B}$ is true $\mathtt{tt}$ (respectively false $\mathtt{ff}$) for the initial environment ρ.

$$\begin{aligned}\mathcal{S}^*[\![\mathtt{S}]\!] \triangleq \quad & \{\langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle \mid \rho \in \mathbb{E}\mathtt{v}\} && \text{(a)}\ (5)\\ & \cup \{\langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle\langle \mathsf{at}[\![\mathtt{S}_t]\!],\ \rho\rangle\pi \mid \mathcal{B}[\![\mathtt{B}]\!]\rho = \mathtt{tt} \wedge \langle \mathsf{at}[\![\mathtt{S}_t]\!],\ \rho\rangle\pi \in \mathcal{S}^*[\![\mathtt{S}_t]\!]\} && \text{(b)}\\ & \cup \{\langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle\langle \mathsf{at}[\![\mathtt{S}_f]\!],\ \rho\rangle\pi \mid \mathcal{B}[\![\mathtt{B}]\!]\rho = \mathtt{ff} \wedge \langle \mathsf{at}[\![\mathtt{S}_f]\!],\ \rho\rangle\pi \in \mathcal{S}^*[\![\mathtt{S}_f]\!]\} && \text{(c)}\end{aligned}$$

Since $\mathsf{brk\text{-}to}[\![\mathtt{S}]\!] = \mathsf{brk\text{-}to}[\![\mathtt{S}_t]\!] = \mathsf{brk\text{-}to}[\![\mathtt{S}_f]\!]$, definitions (5.b) and (5.c) include the cases of breaks respectively from $\mathtt{S}_t$ and $\mathtt{S}_f$ to $\mathsf{brk\text{-}to}[\![\mathtt{S}]\!]$.

- *The prefix trace semantics of the empty statement list* $\mathtt{Sl} = \epsilon$ is reduced to the states at that empty statement list (which is also after that empty statement list).

$$\widehat{\mathscr{S}}^*[\![\mathtt{Sl}]\!] \quad \triangleq \quad \{\langle \mathsf{at}[\![\mathtt{Sl}]\!],\ \rho\rangle \mid \rho \in \mathbb{E}\mathbb{v}\} \tag{6}$$

- The *prefix trace semantics of a non-empty statement list* $\mathtt{Sl} ::= \mathtt{Sl}'\ \mathtt{S}$ are the prefix traces of $\mathtt{Sl}'$ or the finite maximal traces of $\mathtt{Sl}'$ followed by a prefix trace of $\mathtt{S}$.

$$\begin{array}{rcll} \widehat{\mathscr{S}}^*[\![\mathtt{Sl}]\!] & \triangleq & \widehat{\mathscr{S}}^*[\![\mathtt{Sl}']\!] \cup \widehat{\mathscr{S}}^*[\![\mathtt{Sl}']\!] \mathbin{\widehat{;}} \mathscr{S}^*[\![\mathtt{S}]\!] & (7) \\ \mathscr{S} \mathbin{\widehat{;}} \mathscr{S}' & \triangleq & \{\pi \mathbin{\widehat{;}} \pi' \mid \pi \in \mathscr{S} \wedge \pi' \in \mathscr{S}' \wedge \pi \mathbin{\widehat{;}} \pi' \text{ is well-defined}\} & \end{array}$$

Notice that if $\pi \in \widehat{\mathscr{S}}^*[\![\mathtt{Sl}']\!]$, $\pi' \in \mathscr{S}^*[\![\mathtt{S}]\!]$, and $\pi \mathbin{\widehat{;}} \pi' \in \widehat{\mathscr{S}}^*[\![\mathtt{Sl}]\!]$ then the last state of π must be the first state of π' and this state has $\mathsf{at}[\![\mathtt{S}]\!] = \mathsf{aft}[\![\mathtt{Sl}']\!]$ and so the trace π must be a maximal terminating execution of $\mathtt{Sl}'$ *i.e.* $\mathtt{S}$ is executed only if $\mathtt{Sl}'$ terminates.

- *The prefix finite trace semantic definition* $\mathscr{S}^*[\![\mathtt{S}]\!]$ (8) *of an iteration statement of the form* $\mathtt{S} ::= \mathtt{while}^{\,\ell}\ \mathtt{(B)}\ \mathtt{S}_b$ where $\ell = \mathsf{at}[\![\mathtt{S}]\!]$ is the $\subseteq$-least solution $\mathsf{lfp}^{\subseteq} \mathscr{F}^*[\![\mathtt{S}]\!]$ to the equation $X = \mathscr{F}^*[\![\mathtt{S}]\!](X)$. Since $\mathscr{F}^*[\![\mathtt{S}]\!] \in \wp(\mathbb{S}^+) \to \wp(\mathbb{S}^+)$ is $\subseteq$-monotonically increasing (if $X \subseteq X'$ then $\mathscr{F}^*[\![\mathtt{S}]\!](X) \subseteq \mathscr{F}^*[\![\mathtt{S}]\!](X')$) and $\langle \wp(\mathbb{S}^+), \subseteq, \varnothing, \mathbb{S}^+, \cup, \cap\rangle$ is a complete lattice, $\mathsf{lfp}^{\subseteq} \mathscr{F}^*[\![\mathtt{S}]\!]$ exists by Tarski's fixpoint theorem [48] and can be defined as the limit of iterates [15]. In definition (8) of the transformer $\mathscr{F}^*[\![\mathtt{S}]\!]$, case (8.a) corresponds to a loop execution observation stopping on entry, (8.b) corresponds to an observation of a loop exiting after 0 or more iterations, and (8.c) corresponds to a loop execution observation that stops anywhere in the body $\mathtt{S}_b$ after 0 or more iterations. This last case covers the case of an iteration terminated by a break statement (to $\mathsf{aft}[\![\mathtt{S}]\!]$ after the iteration statement).

$$\mathscr{S}^*[\![\mathtt{while}^{\,\ell}\ \mathtt{(B)}\ \mathtt{S}_b]\!] \quad = \quad \mathsf{lfp}^{\subseteq} \mathscr{F}^*[\![\mathtt{while}^{\,\ell}\ \mathtt{(B)}\ \mathtt{S}_b]\!] \tag{8}$$

$$\begin{array}{rcll} \mathscr{F}^*_{\mathbb{S}}[\![\mathtt{while}^{\,\ell}\ \mathtt{(B)}\ \mathtt{S}_b]\!]\, X & \triangleq & \{\langle \ell,\ \rho\rangle \mid \rho \in \mathbb{E}\mathbb{v}\} & \text{(a)} \\ & & \cup \{\pi_2\langle \ell',\ \rho\rangle\langle \mathsf{aft}[\![\mathtt{S}]\!],\ \rho\rangle \mid \pi_2\langle \ell',\ \rho\rangle \in X \wedge \mathscr{B}[\![\mathtt{B}]\!]\,\rho = \mathtt{ff} \wedge \ell' = \ell\}^3 & \text{(b)} \\ & & \cup \{\pi_2\langle \ell',\ \rho\rangle\langle \mathsf{at}[\![\mathtt{S}_b]\!],\ \rho\rangle \cdot \pi_3 \mid \pi_2\langle \ell',\ \rho\rangle \in X \wedge \mathscr{B}[\![\mathtt{B}]\!]\,\rho = \mathtt{tt} \wedge & \text{(c)} \\ & & \qquad \langle \mathsf{at}[\![\mathtt{S}_b]\!],\ \rho\rangle \cdot \pi_3 \in \mathscr{S}^*[\![\mathtt{S}_b]\!] \wedge \ell' = \ell\} & \end{array}$$

- The prefix traces of a program $\mathtt{P} ::= \mathtt{Sl}\ \ell$ are those of the statement list $\mathtt{Sl}$.

$$\mathscr{S}^*[\![\mathtt{P}]\!] \quad \triangleq \quad \mathscr{S}^*[\![\mathtt{Sl}]\!] \tag{9}$$

- The prefix traces of a skip statement $\mathtt{S} ::= \ell\,\mathtt{;}$ where $\ell = \mathsf{at}[\![\mathtt{S}]\!]$ either stop at ℓ are just continue after the skip statement.

$$\mathscr{S}^*[\![\mathtt{S}]\!] \quad \triangleq \quad \{\langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle \mid \rho \in \mathbb{E}\mathbb{v}\} \cup \{\langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle\langle \mathsf{aft}[\![\mathtt{S}]\!],\ \rho\rangle \mid \rho \in \mathbb{E}\mathbb{v}\} \tag{10}$$

- The prefix trace of a compound statement $\mathtt{S} ::= \mathtt{\{\ Sl\ \}}$ are those of its statement list $\mathtt{Sl}$.

$$\mathscr{S}^*[\![\mathtt{S}]\!] \quad \triangleq \quad \mathscr{S}^*[\![\mathtt{Sl}]\!] \tag{11}$$

[3] A definition of the form $d(\vec{x}) \triangleq \{f(\vec{x}') \mid P(\vec{x}', \vec{x})\}$ has the variables $\vec{x}'$ in $P(\vec{x}', \vec{x})$ bound to those of $f(\vec{x}')$ whereas $\vec{x}$ is free in $P(\vec{x}', \vec{x})$ since it appears neither in $f(\vec{x}')$ nor (by assumption) under quantifiers in $P(\vec{x}', \vec{x})$. The $\vec{x}$ of $P(\vec{x}', \vec{x})$ is therefore bound to the $\vec{x}$ of $d(\vec{x})$.

2.5. Semantic properties

As usual in abstract interpretation [16], we represent properties of entities in a universe $\mathbb{U}$ by a subset of this universe. So a property of elements of $\mathbb{U}$ belongs to $\wp(\mathbb{U})$. For example "to be a natural" is the property $\mathbb{N} \triangleq \{n \in \mathbb{Z} \mid n \geqslant 0\}$ of the integers $\mathbb{Z}$. The property "n is a natural" is "$n \in \mathbb{N}$". By program component (safety) property, we understand a property of their prefix trace semantics $\mathcal{S}^*[\![\mathtt{S}]\!] \in \wp(\mathbb{S}^+)$. So program properties P belong to $\wp(\wp(\mathbb{S}^+))$. The *collecting semantics* is the strongest program property, that is the singleton $\{\mathcal{S}^*[\![\mathtt{S}]\!]\}$. The abstraction of $P \in \wp(\wp(\mathbb{S}^+))$ into trace properties in $\wp(\mathbb{S}^+)$ is $\alpha^{\cup}(P) \triangleq \bigcup P$ so for the collecting semantics $\alpha^{\cup}(\{\mathcal{S}^*[\![\mathtt{S}]\!]\}) = \mathcal{S}^*[\![\mathtt{S}]\!]$. In the following, the semantics properties that we consider are trace properties only, in $\alpha^{\cup}(\wp(\wp(\mathbb{S}^+))) = \wp(\mathbb{S}^+)$. So the strongest trace property of a program component $\mathtt{S}$ is its semantics $\mathcal{S}^*[\![\mathtt{S}]\!]$.

3. Specifying computations by regular expressions

Stephen Cole Kleene introduced regular expressions and finite automata to specify execution traces (called events) of automata (called nerve nets) [33]. Kleene proved in [33] that regular expressions and (non-deterministic) finite automata can describe exactly the same classes of languages (see [45, Ch. 1, Sect. 4]). He noted that not all computable execution traces of nerve nets can be (exactly) represented by a regular expression. The situation is the same for programs for which regular expressions (or equivalently finite automata) can specify a super-set of the prefix state trace semantics $\mathcal{S}^*[\![\mathtt{S}]\!]$ of program components $\mathtt{S} \in \mathbb{P}\mathfrak{c}$. An early example is the specification with finite automata specifications for testing [28, 27]. A similar example is security monitors.

Example 3 (Security monitors). Fred Schneider's security monitors [46, 38] use a finite automata specification to state requirements of hardware or software systems. They have been used to check for safety program properties at runtime, that is for any given execution trace in the semantics $\mathcal{S}^*[\![\mathtt{S}]\!]$. The safety property specified by the finite automaton or, equivalently, a regular expression is temporal *i.e.* links events occurring at different times in the computation (such as a file must be opened before being accessed and must eventually be closed). Checking this safety temporal specification for all possible execution traces in the semantics $\mathcal{S}^*[\![\mathtt{S}]\!]$ is a static analysis problem, more precisely, a model checking problem [47], since model checking specializes in temporal properties. □

The use of regular expressions is one distinctive feature of this paper. Regular expressions are commonly used in text editing and understood by all programmers, which is not the case for temporal logics. This is the reason why we use regular expressions as specification language. This also allows us to construct a new model checking algorithm, which would not have been the case with classical temporal logic specifications.

3.1. Syntax of regular expressions

Classical regular expressions denote sets of strings using constants (empty string ε, literal characters a, b, *etc.*) and operator symbols (concatenation $\bullet$, alternation $\mid$, repetition zero or more times * or one or more times $^+$). We replace the literal characters by invariant specifications $\mathtt{L} : \mathtt{B}$ stating that boolean expression $\mathtt{B}$ should be true whenever control reaches any program point in the set $\mathtt{L}$ of program labels. The boolean expression $\mathtt{B}$ may depend on program variables $\mathtt{x}, \mathtt{y}, \ldots \in \mathcal{X}$ and their initial values denoted $\underline{\mathtt{x}}, \underline{\mathtt{y}}, \ldots \in \underline{\mathcal{X}}$ where $\underline{\mathcal{X}} \triangleq \{\underline{\mathtt{x}} \mid \mathtt{x} \in \mathcal{X}\}$.

$$
\begin{array}{rcll}
\mathtt{L} & \in & \wp(\mathbb{L}) & \text{sets of program labels} \\
\mathtt{x}, \mathtt{y}, \ldots & \in & \mathcal{X} & \text{program variables} \\
\underline{\mathtt{x}}, \underline{\mathtt{y}}, \ldots & \in & \underline{\mathcal{X}} & \text{initial values of variables} \\
\mathtt{B} & \in & \mathbb{B} & \text{boolean expressions such that } \mathbb{vars}[\![\mathtt{B}]\!] \subseteq \mathcal{X} \cup \underline{\mathcal{X}}
\end{array}
$$

$$
\begin{array}{rcll}
\mathtt{R} & \in & \mathbb{R} & \text{regular expressions} \quad (12) \\
\mathtt{R} & ::= & \varepsilon & \text{empty} \\
 & \mid & \mathtt{L} : \mathtt{B} & \text{invariant } \mathtt{B} \text{ at } \mathtt{L} \\
 & \mid & \mathtt{R}_1\mathtt{R}_2 \quad (\text{or } \mathtt{R}_1 \bullet \mathtt{R}_2) & \text{concatenation} \\
 & \mid & \mathtt{R}_1 \mid \mathtt{R}_2 & \text{alternative} \\
 & \mid & \mathtt{R}_1^* \quad \mid \quad \mathtt{R}_1^+ & \text{zero/one or more occurrences of } \mathtt{R} \\
 & \mid & (\mathtt{R}_1) & \text{grouping}
\end{array}
$$

We use abbreviations to designate sets of labels such as $\mathtt{?} : \mathtt{B} \triangleq \mathbb{L} : \mathtt{B}$ so that $\mathtt{B}$ is invariant, $\ell : \mathtt{B} \triangleq \{\ell\} : \mathtt{B}$ so that $\mathtt{B}$ is invariant at program label ℓ, $\neg\ell : \mathtt{B} \triangleq \mathbb{L} \setminus \{\ell\} : \mathtt{B}$ when $\mathtt{B}$ holds everywhere but at program point ℓ, *etc.*

Example 4. $(\mathtt{?} : \mathtt{tt})^*$ holds for any program. $(\mathtt{?} : \mathtt{x} >= \mathtt{0})^*$ states that the value of $\mathtt{x}$ is always positive or zero during program execution. $(\mathtt{?} : \mathtt{x} >= \underline{\mathtt{x}})^*$ states that the value of $\mathtt{x}$ is always greater than or equal to its initial value $\underline{\mathtt{x}}$ during execution. $(\mathtt{?} : \mathtt{x} >= \mathtt{0})^* \bullet \ell : \mathtt{x} == \mathtt{0} \bullet (\mathtt{?} : \mathtt{x} < \mathtt{0})^*$ states that the value of $\mathtt{x}$ should be positive or zero and if program point ℓ is ever reached then $\mathtt{x}$ should be 0, and if computations go on after program point ℓ then $\mathtt{x}$ should be negative afterwards. □

Example 5. Continuing Ex. 3 for security monitors, the basic regular expressions are names a of program actions. We can understand such an action a as designating the set $\mathtt{L}$ of labels of all its occurrences in the program. If necessary, the boolean expression $\mathtt{B}$ can be used to specify the parameters of the action. □

There are many regular expressions denoting the language $\{\mathfrak{d}\}$ containing only the empty sequence denoted $\mathfrak{d}$ (such that ε, $\varepsilon\varepsilon$, ε^*, *etc.*), as shown by the following grammar.

$$
\begin{array}{rcll}
\mathtt{R} & \in & \mathbb{R}_\varepsilon & \text{empty regular expressions} \\
\mathtt{R} & ::= & \varepsilon \mid \mathtt{R}_1\mathtt{R}_2 \mid \mathtt{R}_1 \mid \mathtt{R}_2 \mid \mathtt{R}_1^* \mid \mathtt{R}_1^+ \mid (\mathtt{R}_1) & (13)
\end{array}
$$

For specification we use only non-empty regular expressions $\mathtt{R} \in \mathbb{R}^+$ since traces cannot be empty. The definition takes into account the fact that *e.g.* $\mathtt{L : B}$ can be written as $\varepsilon\varepsilon\varepsilon \ldots \varepsilon \mathtt{L : B} \varepsilon\varepsilon \ldots \varepsilon$.

$$\mathtt{R} \in \mathbb{R}^+ \qquad \text{non-empty regular expressions}$$
$$\mathtt{R} ::= \mathtt{L : B} \mid \varepsilon \mathtt{R}_2 \mid \mathtt{R}_1 \varepsilon \mid \mathtt{R}_1 \mathtt{R}_2 \mid \mathtt{R}_1 \mid \mathtt{R}_2 \mid \mathtt{R}_1^+ \mid (\mathtt{R}_1)$$

We also have to consider regular expressions $\mathtt{R} \in \mathbb{R}^\downarrow$ containing no alternative $\mid$.

$$\mathtt{R} \in \mathbb{R}^\downarrow \qquad \mid\text{-free regular expressions}$$
$$\mathtt{R} ::= \varepsilon \mid \mathtt{L : B} \mid \mathtt{R}_1 \mathtt{R}_2 \mid \mathtt{R}_1^* \mid \mathtt{R}_1^+ \mid (\mathtt{R}_1)$$

3.2. Relational semantics of regular expressions

The semantics (2) of expressions is changed as follows ($\underline{\varrho}(\mathtt{x})$ denotes the initial values $\underline{\mathtt{x}}$ of variables $\mathtt{x}$ and $\rho(\mathtt{x})$ their current value, $\uparrow$ is the alternative denial logical operation)

$$\begin{array}{llll} \mathscr{A}[\![\mathtt{1}]\!]\underline{\varrho}, \rho \triangleq 1 & & \mathscr{A}[\![\mathtt{A}_1 - \mathtt{A}_2]\!]\underline{\varrho}, \rho \triangleq \mathscr{A}[\![\mathtt{A}_1]\!]\underline{\varrho}, \rho - \mathscr{A}[\![\mathtt{A}_2]\!]\underline{\varrho}, \rho \\ \mathscr{A}[\![\underline{\mathtt{x}}]\!]\underline{\varrho}, \rho \triangleq \underline{\varrho}(\mathtt{x}) & & \mathscr{B}[\![\mathtt{A}_1 < \mathtt{A}_2]\!]\underline{\varrho}, \rho \triangleq \mathscr{A}[\![\mathtt{A}_1]\!]\underline{\varrho}, \rho < \mathscr{A}[\![\mathtt{A}_2]\!]\underline{\varrho}, \rho \\ \mathscr{A}[\![\mathtt{x}]\!]\underline{\varrho}, \rho \triangleq \rho(\mathtt{x}) & & \mathscr{B}[\![\mathtt{B}_1 \text{ nand } \mathtt{B}_2]\!]\underline{\varrho}, \rho \triangleq \mathscr{B}[\![\mathtt{B}_1]\!]\underline{\varrho}, \rho \uparrow \mathscr{B}[\![\mathtt{B}_2]\!]\underline{\varrho}, \rho \end{array} \tag{14}$$

We represent a non-empty finite sequence $\sigma_1 \ldots \sigma_n \in \mathbb{S}^+ \triangleq \bigcup_{n \in \mathbb{N} \setminus \{0\}} [1, n] \to \mathbb{S}$ of states $\sigma_i \in \mathbb{S} \triangleq (\mathbb{L} \times \mathbb{E}\mathtt{v})$ by a map $\sigma \in [1, n] \to \mathbb{S}$ (which is the empty sequence $\sigma = \mathfrak{d}$ when $n = 0$).

The relational semantics $\mathscr{S}^{\mathtt{r}}[\![\mathtt{R}]\!] \in \wp(\mathbb{E}\mathtt{v} \times \mathbb{S}^*)$ of regular expressions $\mathtt{R}$ relates an arbitrary initial environment $\underline{\varrho} \in \mathbb{E}\mathtt{v}$ to a trace $\pi \in \mathbb{S}^*$ by defining how the states of the trace π are related to that initial environment $\underline{\varrho}$.

$$\begin{array}{ll} \mathscr{S}^{\mathtt{r}}[\![\varepsilon]\!] \triangleq \{\langle \underline{\varrho}, \mathfrak{d} \rangle \mid \underline{\varrho} \in \mathbb{E}\mathtt{v}\} & \mathscr{S}^{\mathtt{r}}[\![\mathtt{R}]\!]^0 \triangleq \mathscr{S}^{\mathtt{r}}[\![\varepsilon]\!] \\ \mathscr{S}^{\mathtt{r}}[\![\mathtt{L : B}]\!] \triangleq \{\langle \underline{\varrho}, \langle \ell, \rho \rangle \rangle \mid \ell \in \mathtt{L} \wedge \mathscr{B}[\![\mathtt{B}]\!]\underline{\varrho}, \rho\} & \mathscr{S}^{\mathtt{r}}[\![\mathtt{R}]\!]^{n+1} \triangleq \mathscr{S}^{\mathtt{r}}[\![\mathtt{R}]\!]^n \circledast \mathscr{S}^{\mathtt{r}}[\![\mathtt{R}]\!] \\ \mathscr{S}^{\mathtt{r}}[\![\mathtt{R}_1 \mathtt{R}_2]\!] \triangleq \mathscr{S}^{\mathtt{r}}[\![\mathtt{R}_1]\!] \circledast \mathscr{S}^{\mathtt{r}}[\![\mathtt{R}_2]\!] & \mathscr{S}^{\mathtt{r}}[\![\mathtt{R}^*]\!] \triangleq \bigcup_{n \in \mathbb{N}} \mathscr{S}^{\mathtt{r}}[\![\mathtt{R}]\!]^n \\ \mathscr{S} \circledast \mathscr{S}' \triangleq \{\langle \underline{\varrho}, \pi \cdot \pi' \rangle \mid \langle \underline{\varrho}, \pi \rangle \in \mathscr{S} \wedge \langle \underline{\varrho}, \pi' \rangle \in \mathscr{S}'\} & \mathscr{S}^{\mathtt{r}}[\![\mathtt{R}^+]\!] \triangleq \bigcup_{n \in \mathbb{N} \setminus \{0\}} \mathscr{S}^{\mathtt{r}}[\![\mathtt{R}]\!]^n \\ \mathscr{S}^{\mathtt{r}}[\![\mathtt{R}_1 \mid \mathtt{R}_2]\!] \triangleq \mathscr{S}^{\mathtt{r}}[\![\mathtt{R}_1]\!] \cup \mathscr{S}^{\mathtt{r}}[\![\mathtt{R}_2]\!] & \mathscr{S}^{\mathtt{r}}[\![(\mathtt{R})]\!] \triangleq \mathscr{S}^{\mathtt{r}}[\![\mathtt{R}]\!] \end{array} \tag{15}$$

Example 6. The semantics of the regular expression $\mathtt{R} \triangleq \ell : \mathtt{x} = \underline{\mathtt{x}} \bullet \ell' : \mathtt{x} = \underline{\mathtt{x}} + 1$ is $\mathscr{S}^{\mathtt{r}}[\![\mathtt{R}]\!] = \{\langle \underline{\varrho}, \langle \ell, \rho \rangle \langle \ell', \rho' \rangle \rangle \mid \rho(\mathtt{x}) = \underline{\varrho}(\mathtt{x}) \wedge \rho'(\mathtt{x}) = \underline{\varrho}(\mathtt{x}) + 1\}$ meaning that if the initial value of $\mathtt{x}$ is $\underline{\mathtt{x}}$ at ℓ then, after one step of computation, it is $\underline{\mathtt{x}} + 1$ at ℓ'.[4] □

[4]The trace semantics for regular expressions always relates variable environments to the initial environment, so that one cannot express sequences of variable increments. A possible refined specification for an extension of this paper would allow for recording intermediate or final values of variables during computations, not only the initial ones.

4. Definition of regular model checking

Let the prefix closure $\mathsf{prefix}(\Pi)$ of a set $\Pi \in \wp(\mathbb{E}\mathsf{v} \times \mathbb{S}^+)$ of traces be

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

The following Def. 7 defines the model checking problem $\mathtt{P}, \underline{\varrho} \vDash \mathtt{R}$ as checking that the semantics of the given program $\mathtt{P} \in \mathbb{P}$ meets the regular[5] specification $\mathtt{R} \in \mathbb{R}^+$ for the initial environment $\underline{\varrho}$.

Definition 7 (Model checking).

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

The prefix closure prefix allows the regular specification $\mathtt{R}$ to specify traces satisfying a prefix of the specification only, as in $\ell\ \mathtt{x = x + 1\ ;}\ell', \underline{\varrho} \vDash \mathtt{R}$ where $\mathtt{R} = \ell : \mathtt{x} = \underline{\mathtt{x}} \bullet \ell' : \mathtt{x} = \underline{\mathtt{x}} + 1 \bullet \ell'' : \mathtt{x} = \underline{\mathtt{x}} + 3$ and $\underline{\varrho}(\mathtt{x}) = \underline{\mathtt{x}}$.

The extension of the specification by $(\mathtt{?} : \mathtt{tt})^*$ allows for the regular specification $\mathtt{R}$ to specify only a prefix of the traces, such as $\ell\ \mathtt{x = x + 1\ ;}\ell'\ \mathtt{x = x + 2\ ;}\ell'', \underline{\varrho} \vDash \ell : \mathtt{x} = \underline{\mathtt{x}} \bullet \ell' : \mathtt{x} = \underline{\mathtt{x}} + 1$.

Model checking is a boolean abstraction $\langle \wp(\mathbb{S}^+), \subseteq\rangle \underset{\alpha_{\underline{\varrho},\mathtt{R}}}{\overset{\gamma_{\underline{\varrho},\mathtt{R}}}{\leftrightarrows}} \langle \mathbb{B}, \Leftarrow\rangle$ where $\alpha_{\underline{\varrho},\mathtt{R}}(\Pi) \triangleq (\{\underline{\varrho}\} \times \Pi) \subseteq \mathsf{prefix}(\mathcal{S}^r[\![\mathtt{R} \bullet (\mathtt{?} : \mathtt{tt})^*]\!]))$.

5. Properties of regular expressions

We recall properties of regular expressions that we will use to design a structural model checking abstract semantics.

5.1. Equivalence of regular expressions

We say that regular expressions are equivalent when they have the same semantics.

$$\mathtt{R}_1 \eqsim \mathtt{R}_2 \triangleq (\mathcal{S}^r[\![\mathtt{R}_1]\!] = \mathcal{S}^r[\![\mathtt{R}_2]\!])$$

5.2. Disjunctive normal form dnf *of regular expressions*

As noticed by Kleene [33, p. 14], regular expressions can be put in the equivalent disjunctive normal form of Hilbert—Ackermann [29, Ch I, § 3 and Ch III, § 8]. 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

[5] We understand "regular model checking" as checking temporal specifications given by a regular expression. This is different from [1] model checking transition systems which states are regular word or tree languages.

occurrence of $\mid$. Any regular expression R has a disjunctive normal form $\mathsf{dnf}(\mathsf{R})$ defined as follows.

$$\begin{array}{rcl}
\mathsf{dnf}(\varepsilon) & \triangleq & \varepsilon \\
\mathsf{dnf}(\mathsf{L} : \mathsf{B}) & \triangleq & \mathsf{L} : \mathsf{B} \\
\mathsf{dnf}(\mathsf{R}_1\mathsf{R}_2) & \triangleq & \mathsf{let}\ \mathsf{R}_1^1 \mid \ldots \mid \mathsf{R}_1^{n_1} = \mathsf{dnf}(\mathsf{R}_1)\ \mathsf{and}\ \mathsf{R}_2^1 \mid \ldots \mid \mathsf{R}_2^{n_2} = \mathsf{dnf}(\mathsf{R}_2)\ \mathsf{in} \bigmid_{i=1}^{n_1} \bigmid_{j=1}^{n_2} \mathsf{R}_1^i\mathsf{R}_2^j \\
\mathsf{dnf}(\mathsf{R}_1 \mid \mathsf{R}_2) & \triangleq & \mathsf{dnf}(\mathsf{R}_1) \mid \mathsf{dnf}(\mathsf{R}_2) \\
\mathsf{dnf}(\mathsf{R}^*) & \triangleq & \mathsf{let}\ \mathsf{R}^1 \mid \ldots \mid \mathsf{R}^n = \mathsf{dnf}(\mathsf{R})\ \mathsf{in}\ ((\mathsf{R}^1)^* \ldots (\mathsf{R}^n)^*)^* \\
\mathsf{dnf}(\mathsf{R}^+) & \triangleq & \mathsf{dnf}(\mathsf{R}\mathsf{R}^*) \\
\mathsf{dnf}((\mathsf{R})) & \triangleq & (\mathsf{dnf}(\mathsf{R}))
\end{array} \tag{16}$$

The proof that $\mathsf{dnf}(\mathsf{R})$ is in disjunctive normal form is by structural induction using the fact that the $\mathsf{R}^1, \ldots, \mathsf{R}^n$ do not contain any $\mid$.

The Lem. 8 below shows that normalization leaves the semantics unchanged. It uses the fact that $(\mathsf{R}_1 \mid \mathsf{R}_2)^* \eqsim (\mathsf{R}_1{}^*\mathsf{R}_2{}^*)^*$ where the R_1 and R_2 do not contain any $\mid$ [31, Sect. 3.4.6, p. 118].

Lemma 8. $\mathsf{dnf}(\mathsf{R}) \eqsim \mathsf{R}$.

Lem. 8 shows that normalization in (16) can be further simplified by $\varepsilon\mathsf{R} \eqsim \mathsf{R}\varepsilon \eqsim \mathsf{R}$ and $(\varepsilon)^* \eqsim \varepsilon$ which have equivalent semantics.

5.3. first *and* next *of regular expressions*

Janusz Brzozowski [7] introduced the notion of derivation for regular expressions (extended with arbitrary Boolean operations[6]). The derivative of a regular expression R with respect to a symbol a, typically denoted as $D_a(\mathsf{R})$ or $a^{-1}\mathsf{R}$, is a regular expression given by a simple recursive definition on the syntactic structure of R. The crucial property of these derivatives is that a string of the form $a\sigma$ (starting with the symbol a) matches an expression R iff the suffix σ matches the derivative $D_a(\mathsf{R})$ [7, 41, 2].

Following this idea, assume that a non-empty regular expression $\mathsf{R} \in \mathbb{R}^+$ has been decomposed into disjunctive normal form $(\mathsf{R}_1 \mid \ldots \mid \mathsf{R}_n)$ for some $n \geqslant 1$, in which none of the R_i, for $i \in [1, n]$, contains an occurrence of $\mid$. We can further decompose each $\mathsf{R}_i \in \mathbb{R}^+ \cap \mathbb{R}^\downarrow$ into $\langle \mathsf{L} : \mathsf{B},\ \mathsf{R}_i' \rangle = \mathsf{fstnxt}(\mathsf{R}_i)$ such that

- $\mathsf{L} : \mathsf{B}$ recognizes the first state of sequences of states recognized by R_i;
- the regular expression R_i' recognizes sequences of states after the first state of sequences of states recognized by R_i.

We define fstnxt for non-empty $\mid$-free regular expressions $\mathsf{R} \in \mathbb{R}^+ \cap \mathbb{R}^\downarrow$ by structural induction, as follows.

[6] which would be a possible refined specification for an extension of this paper.

$$
\begin{array}{rcll}
\mathsf{fstnxt}(\mathtt{L : B}) & \triangleq & \langle \mathtt{L : B},\, \varepsilon \rangle & (17)\\
\mathsf{fstnxt}(\mathtt{R}_1\mathtt{R}_2) & \triangleq & \mathsf{fstnxt}(\mathtt{R}_2) & \text{if } \mathtt{R}_1 \in \mathbb{R}_\varepsilon \\
\mathsf{fstnxt}(\mathtt{R}_1\mathtt{R}_2) & \triangleq & \mathsf{let}\ \langle \mathtt{R}_1^f,\, \mathtt{R}_1^n \rangle = \mathsf{fstnxt}(\mathtt{R}_1)\ \mathsf{in}\ [\![\, \mathtt{R}_1^n \in \mathbb{R}_\varepsilon\ ?\ \langle \mathtt{R}_1^f,\, \mathtt{R}_2 \rangle\ :\ \langle \mathtt{R}_1^f,\, \mathtt{R}_1^n \bullet \mathtt{R}_2 \rangle \,]\!] & \\
 & & & \text{if } \mathtt{R}_1 \notin \mathbb{R}_\varepsilon \\
\mathsf{fstnxt}(\mathtt{R}^+) & \triangleq & \mathsf{let}\ \langle \mathtt{R}^f,\, \mathtt{R}^n \rangle = \mathsf{fstnxt}(\mathtt{R})\ \mathsf{in}\ [\![\, \mathtt{R}^n \in \mathbb{R}_\varepsilon\ ?\ \langle \mathtt{R}^f,\, \mathtt{R}^* \rangle\ :\ \langle \mathtt{R}^f,\, \mathtt{R}^n \bullet \mathtt{R}^* \rangle \,]\!] & \\
\mathsf{fstnxt}((\mathtt{R})) & \triangleq & \mathsf{fstnxt}(\mathtt{R}) &
\end{array}
$$

The following Lem. 9 shows the equivalence of an alternative-free regular expression and its first and next decomposition.

Lemma 9. Let $\mathtt{R} \in \mathbb{R}^+ \cap \mathbb{R}^{\mid}$ be a non-empty $\mid$-free regular expression and $\langle \mathtt{L : B},\, \mathtt{R}' \rangle = \mathsf{fstnxt}(\mathtt{R})$. Then $\mathtt{R}' \in \mathbb{R}^{\mid}$ is $\mid$-free and $\mathtt{R} \eqsim \mathtt{L : B} \bullet \mathtt{R}'$.

6. The model checking abstraction

The model checking abstraction in Section **4** is impractical for structural model checking. Assume, for example, that we check a trace concatenation $\pi_1 \cdot \pi_2$ of a statement list $\mathtt{Sl} ::= \mathtt{Sl}'\ \mathtt{S}$ for a specification $\mathtt{R}$ where π_1 is a trace of $\mathtt{Sl}'$ and π_2 is a trace of $\mathtt{S}$. We first check that π_1 satisfies $\mathtt{R}$. We must then check π_2 for a continuation $\mathtt{R}_2$ of $\mathtt{R}$ which should be derived from π_1 and $\mathtt{R}$. This continuation is not provided by the boolean abstraction $\alpha_{\underline{\varrho},\mathtt{R}}$ which needs to be refined as shown below.

Example 10. Assume we want to check $\ell_1\ \mathtt{x = x + 1\ ;}\ell_2\ \mathtt{x = x + 2\ ;}\ell_3$ for the regular specification $\mathtt{? : x} = \underline{\mathtt{x}} \bullet \mathtt{? : x} = \underline{\mathtt{x}} + 1 \bullet \mathtt{? : x} = \underline{\mathtt{x}} + 3$ by first checking the first statement and then the second. Knowing the boolean information that $\ell_1\ \mathtt{x = x + 1\ ;}\ell_2$ model checks for $\mathtt{? : x} = \underline{\mathtt{x}} \bullet \mathtt{? : x} = \underline{\mathtt{x}} + 1$ is not enough. We must also know what to check the continuation $\ell_2\ \mathtt{x = x + 2\ ;}\ell_3$ for. (This is $\mathtt{? : x} = \underline{\mathtt{x}} + 1 \bullet \mathtt{? : x} = \underline{\mathtt{x}} + 3$ that is if $\mathtt{x}$ is equal to the initial value plus 1 at ℓ_2, it is equal to this initial value plus 3 at ℓ_3.) □

The model-checking $\mathscr{M}^t\langle \underline{\varrho},\, \mathtt{R} \rangle \pi$ of a trace π with initial environment $\underline{\varrho}$ for a $\mid$-free specification $\mathtt{R} \in \mathbb{R}^{\mid}$ is a pair $\langle b,\, \mathtt{R}' \rangle$ where the boolean b states whether the specification $\mathtt{R}$ holds for the trace π and $\mathtt{R}'$ specifies the possible continuations of π according to $\mathtt{R}$, ε if none.

Example 11. For $\mathtt{Sl} = \ell_1\ \mathtt{x = x + 1\ ;}\ell_2\ \mathtt{x = x + 2\ ;}\ell_3$, we have $\mathscr{S}^*[\![\mathtt{Sl}]\!] = \{\langle \ell_1,\, \rho \rangle \langle \ell_2,\, \rho[\mathtt{x} \leftarrow \rho(\mathtt{x}) + 1] \rangle \langle \ell_3,\, \rho[\mathtt{x} \leftarrow \rho(\mathtt{x}) + 3] \rangle \mid \rho \in \mathbb{E}\mathtt{v}\}$ and $\mathscr{M}^t\langle \rho,\, \mathtt{? : x} = \underline{\mathtt{x}} \bullet \mathtt{? : x} = \underline{\mathtt{x}} + 1 \bullet \mathtt{? : x} = \underline{\mathtt{x}} + 3 \rangle (\langle \ell_1,\, \rho \rangle \langle \ell_2,\, \rho[\mathtt{x} \leftarrow \rho(\mathtt{x}) + 1] \rangle \langle \ell_3,\, \rho[\mathtt{x} \leftarrow \rho(\mathtt{x}) + 3] \rangle) = \langle \mathtt{tt},\, \varepsilon \rangle$ (we have ignored the initial empty statement list in $\mathtt{Sl}$ to simplify the specification). □

The fact that $\mathscr{M}^t\langle \underline{\varrho},\, \mathtt{R} \rangle \pi$ returns a pair $\langle b,\, \mathtt{R}' \rangle$ where $\mathtt{R}'$ is to be satisfied by continuations of π allows us to perform program model checking by structural induction on the program in Section **8**. The formal definition is the following.

Definition 12 (Regular model checking).

- Trace model checking ($\underline{\varrho} \in \mathbb{E}\mathsf{v}$ is an initial environment and $\mathtt{R} \in \mathscr{R}^{+} \cap \mathscr{R}^{\nmid}$ is a non-empty and $\mid$-free regular expression):

$$\begin{array}{lcll} \mathscr{M}^{t}\langle \underline{\varrho},\ \varepsilon\rangle\pi & \triangleq & \langle \mathtt{tt},\ \varepsilon\rangle & (18)\\ \mathscr{M}^{t}\langle \underline{\varrho},\ \mathtt{R}\rangle\ni & \triangleq & \langle \mathtt{tt},\ \mathtt{R}\rangle & \\ \mathscr{M}^{t}\langle \underline{\varrho},\ \mathtt{R}\rangle\pi & \triangleq & \mathsf{let}\ \langle \ell_1,\ \rho_1\rangle\pi' = \pi\ \mathsf{and}\ \langle \mathtt{L}\ \mathtt{:}\ \mathtt{B},\ \mathtt{R}'\rangle = \mathsf{fstnxt}(\mathtt{R})\ \mathsf{in} \quad \pi \neq \ni & \\ & & \quad [\![\ \langle \underline{\varrho},\ \langle \ell_1,\ \rho_1\rangle\rangle \in \mathscr{S}^{\mathsf{r}}[\![\mathtt{L}\ \mathtt{:}\ \mathtt{B}]\!]\ ?\ \mathscr{M}^{t}\langle \underline{\varrho},\ \mathtt{R}'\rangle\pi'\ \wr\ \langle \mathtt{ff},\ \mathtt{R}'\rangle\ ]\!] & \end{array}$$

- Set of traces model checking (for an $\mid$-free regular expression $\mathtt{R} \in \mathscr{R}^{\nmid}$):

$$\mathscr{M}^{\nmid}\langle \underline{\varrho},\ \mathtt{R}\rangle\Pi \triangleq \{\langle \pi,\ \mathtt{R}'\rangle \mid \pi \in \Pi \wedge \langle \mathtt{tt},\ \mathtt{R}'\rangle = \mathscr{M}^{t}\langle \underline{\varrho},\ \mathtt{R}\rangle\pi\} \quad (19)$$

- Program component $\mathtt{S} \in \mathscr{P}\mathscr{C}$ model checking (for an $\mid$-free regular expression $\mathtt{R} \in \mathscr{R}^{\nmid}$):

$$\mathscr{M}^{\nmid}[\![\mathtt{S}]\!]\langle \underline{\varrho},\ \mathtt{R}\rangle \triangleq \mathscr{M}^{\nmid}\langle \underline{\varrho},\ \mathtt{R}\rangle(\mathscr{S}^{*}[\![\mathtt{S}]\!]) \quad (20)$$

- Set of traces model checking (for regular expression $\mathtt{R} \in \mathscr{R}$):

$$\begin{array}{lcll} \mathscr{M}\langle \underline{\varrho},\ \mathtt{R}\rangle\Pi & \triangleq & \mathsf{let}\ (\mathtt{R}_1 \mid \ldots \mid \mathtt{R}_n) = \mathsf{dnf}(\mathtt{R})\ \mathsf{in} & (21)\\ & & \quad \bigcup_{i=1}^{n} \{\pi \mid \exists \mathtt{R}' \in \mathscr{R}\ .\ \langle \pi,\ \mathtt{R}'\rangle \in \mathscr{M}^{\nmid}\langle \underline{\varrho},\ \mathtt{R}_i\rangle\Pi\} & \end{array}$$

- Model checking of a program component $\mathtt{S} \in \mathscr{P}\mathscr{C}$ (for regular expression $\mathtt{R} \in \mathscr{R}$):

$$\mathscr{M}[\![\mathtt{S}]\!]\langle \underline{\varrho},\ \mathtt{R}\rangle \triangleq \mathscr{M}\langle \underline{\varrho},\ \mathtt{R}\rangle(\mathscr{S}^{*}[\![\mathtt{S}]\!]) \quad (22) \quad \square$$

The model checking $\mathscr{M}^{t}\langle \underline{\varrho}, \mathtt{R}\rangle\pi$ of a trace π in (18) returns a pair $\langle b, \mathtt{R}'\rangle$ specifying whether π satisfies the specification $\mathtt{R}$ (when $b = \mathtt{tt}$) or not (when $b = \mathtt{ff}$). So if $\mathscr{M}^{t}\langle \rho,\ \mathtt{R}\rangle(\pi) = \langle \mathtt{ff},\ \mathtt{R}'\rangle$ in (19) then the trace π is a counter example to the specification $\mathtt{R}$. $\mathtt{R}'$ specifies what a continuation π' of π would have to satisfy for $\pi \frown \pi'$ to satisfy $\mathtt{R}$ (nothing specific when $\mathtt{R}' = \varepsilon$).

Notice that $\mathscr{M}^{t}\langle \underline{\varrho}, \mathtt{R}\rangle\pi$ checks whether the given trace π satisfies the regular specification $\mathtt{R}$ for initial environment $\underline{\varrho}$. Because only one trace is involved, this check can be done at runtime using a monitoring of the program execution. This is the case for Fred Schneider's security monitors [46] in Ex. 3 (using an equivalent specification by finite automata).

The set of traces model checking $\mathscr{M}^{\nmid}\langle \underline{\varrho}, \mathtt{R}\rangle\Pi$ returns the subset of traces of Π satisfying the specification $\mathtt{R}$ for the initial environment $\underline{\varrho}$. Since all program

executions $\mathcal{S}^*[\![\mathtt{P}]\!]$ are involved, the model checking $\mathcal{M}^\downarrow[\![\mathtt{P}]\!]\langle \underline{\varrho}, \mathtt{R}\rangle$ of a program P becomes, by Rice theorem [44], undecidable.

The regular specification R is relational in that it may relate the initial and current states (or else may only assert a property of the current states when R never refer to the initial environment $\underline{\varrho}$). If $\pi\langle \ell, \rho\rangle\pi' \in \mathcal{S}^*[\![\mathtt{S}]\!]$ is an execution trace satisfying the specification R then R in (22) determines a relationship between the initial environment $\underline{\varrho}$ and the current environment ρ. For example $\mathtt{R} = \langle\{\mathtt{at}[\![\mathtt{S}]\!]\}, \mathtt{B}\rangle \bullet \mathtt{R}'$ with $\mathcal{B}[\![\mathtt{B}]\!]\underline{\varrho}, \rho = \forall \mathtt{x} \in \mathbb{X} \,.\, \underline{\varrho}(\underline{\mathtt{x}}) = \rho(\mathtt{x})$ expresses that the initial values of variables x are denoted $\underline{\mathtt{x}}$. $\mathcal{B}[\![\mathtt{B}]\!]\underline{\varrho}, \rho = \mathtt{tt}$ would state that there is no constraint on the initial value of variables. The difference with the invariant specifications of is that the order of computations is preserved. R can specify in which order program points may be reached, which is impossible with invariants [7].

The model checking abstraction (19) which, given an initial environment $\underline{\varrho} \in \mathbb{E}^\mathbb{V}$ and an $\mathtt{I}$-free regular specification $\mathtt{R} \in \mathcal{R}^\downarrow$, returns the set of traces satisfying this specification is the lower adjoint of the Galois connection

$$\langle \wp(\mathbb{S}^+), \subseteq\rangle \underset{\mathcal{M}^\downarrow\langle \underline{\varrho}, \mathtt{R}\rangle}{\overset{\gamma_{\mathcal{M}^\downarrow\langle \underline{\varrho}, \mathtt{R}\rangle}}{\leftrightarrows}} \langle \wp(\mathbb{S}^+ \times \mathcal{R}^\downarrow), \subseteq\rangle \quad \text{for } \mathtt{R} \in \mathcal{R}^\downarrow \text{ in (19)} \tag{23}$$

If $\langle C, \leqslant\rangle$ is a poset, $\langle \mathcal{A}, \sqsubseteq, \sqcup, \sqcap\rangle$ is a complete lattice, $\forall i \in [1, n] \,.\, \langle C, \leqslant\rangle \underset{\alpha_i}{\overset{\gamma_i}{\leftrightarrows}} \langle \mathcal{A}, \sqsubseteq\rangle$ then $\langle C, \leqslant\rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle \mathcal{A}, \sqsubseteq\rangle$ where $\alpha \triangleq \dot{\bigsqcup_{i=1}^{n}} \alpha_i$ and $\gamma = \dot{\bigsqcap_{i=1}^{n}} \gamma_i$, pointwise. This implies that

$$\langle \wp(\mathbb{S}^+), \subseteq\rangle \underset{\mathcal{M}\langle \underline{\varrho}, \mathtt{R}\rangle}{\overset{\gamma_{\mathcal{M}\langle \underline{\varrho}, \mathtt{R}\rangle}}{\leftrightarrows}} \langle \wp(\mathbb{S}^+), \subseteq\rangle \quad \text{for } \mathtt{R} \in \mathcal{R} \text{ in (21)} \tag{24}$$

To follow the tradition that model checking returns a boolean answer this abstraction can be composed with the boolean abstraction

$$\langle \wp(\mathbb{S}^+), \subseteq\rangle \underset{\alpha_{\mathcal{M}\langle \underline{\varrho}, \mathtt{R}\rangle}}{\overset{\gamma_{\mathcal{M}\langle \underline{\varrho}, \mathtt{R}\rangle}}{\leftrightarrows}} \langle \mathbb{B}, \Leftarrow\rangle \tag{25}$$

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

7. Soundness and completeness of the model checking abstraction

The following Th. 13 shows that the Def. 7 of model checking a program semantics for a regular specification is a sound and complete abstraction of this semantics.

[7] By introduction of an auxiliary variable C incremented at each program step one can simulate a trace with invariants. But then the reasoning is not on the original program P but on a transformed program $\overline{\mathtt{P}}$. Invariants in $\overline{\mathtt{P}}$ holding for a given value of c of C also hold at the position c of the traces in P. This kind of indirect reasoning is usually heavy and painful to maintain when programs are modified since values of counters are no longer the same. The use of temporal specifications has the advantage of avoiding the reasoning on explicit positions in the trace.

Theorem 13 (Model checking soundness (⇐) and completeness (⇒)).

$$\mathsf{P}, \underline{\varrho} \vDash \mathsf{R} \quad \Leftrightarrow \quad \alpha_{\mathscr{M}\langle \underline{\varrho}, \mathsf{R}\rangle}(\mathscr{S}^*[\![\mathsf{P}]\!])$$

We decompose the proof of Th. 13 into soundness (⇐) and completeness (⇒). Soundness directly follows from the following

Lemma 14 (Model checking soundness). $\mathscr{M}[\![\mathsf{P}]\!]\mathsf{R} \subseteq \mathsf{prefix}(\mathscr{S}^r[\![\mathsf{R} \bullet (? : \mathtt{tt})^*]\!])$.

Proof (of Lem. 14). Let $\mathscr{M}^t\langle \underline{\varrho}, \mathsf{R}\rangle\pi$ be the model checking of a prefix trace $\pi = \sigma_1 \dots \sigma_\ell = \langle \ell_1, \rho_1\rangle \dots \langle \ell_\ell, \rho_\ell\rangle$ for an initial environment $\underline{\varrho}$ and a specification $\mathsf{R} \notin \mathbb{R}_\varepsilon$. Consider the repeated unrolling of R by fstnxt, which by Lem. 9, is as follows.

$$\begin{array}{lll} & \mathsf{R} & \\ \eqsim & \mathsf{L}_1 : \mathsf{B}_1 \bullet \mathsf{R}_1 & \mathsf{fstnxt}(\mathsf{R}) \triangleq \langle \mathsf{L}_1 : \mathsf{B}_1, \mathsf{R}_1\rangle \\ \eqsim & \mathsf{L}_1 : \mathsf{B}_1 \bullet \mathsf{L}_2 : \mathsf{B}_2 \bullet \mathsf{R}_2 & \mathsf{fstnxt}(\mathsf{R}_1) \triangleq \langle \mathsf{L}_2 : \mathsf{B}_2, \mathsf{R}_2\rangle \\ & \dots & \\ \eqsim & \mathsf{L}_1 : \mathsf{B}_1 \bullet \mathsf{L}_2 : \mathsf{B}_2 \bullet \dots \bullet \mathsf{L}_k : \mathsf{B}_k \bullet \mathsf{R}_k & \mathsf{fstnxt}(\mathsf{R}_{k-1}) \triangleq \langle \mathsf{L}_k : \mathsf{B}_k, \mathsf{R}_k\rangle \\ & \dots & \end{array}$$

Either this development stops at some $k = n$ with $\mathsf{R}_n = \varepsilon$ or goes on forever, in which case $n = +\infty$. Let us prove that

(14.1) Either the specification R is satisfied, in which case $\mathsf{let}\, m = \min(n, \ell)\ \mathsf{in}\ \forall i \in [1, m]\ .\ \langle \underline{\varrho}, \langle \ell_i, \rho_i\rangle\rangle \in \mathscr{S}^r[\![\mathsf{L}_i : \mathsf{B}_i]\!]$ and so $\mathscr{M}^t\langle \underline{\varrho}, \mathsf{R}\rangle\pi = \langle \mathtt{tt}, \mathsf{R}_m\rangle$;

(14.2) Or the specification R is not satisfied, in which case $\exists i \in [1, \min(n, \ell)]\ .\ \forall j \in [1, i[\ .\ \langle \underline{\varrho}, \langle \ell_j, \rho_j\rangle\rangle \in \mathscr{S}^r[\![\mathsf{L}_j : \mathsf{B}_j]\!] \wedge \langle \underline{\varrho}, \langle \ell_i, \rho_i\rangle\rangle \notin \mathscr{S}^r[\![\mathsf{L}_i : \mathsf{B}_i]\!]$ and so $\mathscr{M}^t\langle \underline{\varrho}, \mathsf{R}\rangle\pi = \langle \mathtt{ff}, \mathsf{R}_i\rangle$.

The proof is by induction on $m = \min(n, \ell)$. We have $n \geqslant 1$ since $\mathsf{R} \notin \mathbb{R}_\varepsilon$ so $\mathsf{fstnxt}(\mathsf{R}) = \langle \mathsf{L}_1 : \mathsf{B}_1, \mathsf{R}_1\rangle$ and $\pi = \langle \ell_1, \rho_1\rangle\pi_1$ since traces in Section **2** are not empty.

— For the basis $n = 1$ or $\ell = 1$, (18) implies $\mathscr{M}^t\langle \underline{\varrho}, \mathsf{R}\rangle\pi = \langle b, \mathsf{R}_1\rangle$ where $b = \langle \underline{\varrho}, \langle \ell_1, \rho_1\rangle\rangle \in \mathscr{S}^r[\![\mathsf{L}_1 : \mathsf{B}_1]\!]$ which is case (14.1) when $b = \mathtt{tt}$ and (14.2) when $b = \mathtt{ff}$.

— For the induction case $m = \min(n, \ell) > 1$.

- Either $\langle \underline{\varrho}, \langle \ell_1, \rho_1\rangle\rangle \notin \mathscr{S}^r[\![\mathsf{L}_1 : \mathsf{B}_1]\!]$ and, by (18), $\mathscr{M}^t\langle \underline{\varrho}, \mathsf{R}\rangle\pi \triangleq \langle \mathtt{ff}, \mathsf{R}_1\rangle$, so we are in case (14.2) with $i = 1$;
- Or $\langle \underline{\varrho}, \langle \ell_1, \rho_1\rangle\rangle \in \mathscr{S}^r[\![\mathsf{L}_1 : \mathsf{B}_1]\!]$ and there are two subcases.
  - If the specification R_1 is satisfied from 2 up to m, in which case $\forall i \in [2, m]\ .\ \langle \underline{\varrho}, \langle \ell_i, \rho_i\rangle\rangle \in \mathscr{S}^r[\![\mathsf{L}_i : \mathsf{B}_i]\!]$ and so $\mathscr{M}^t\langle \underline{\varrho}, \mathsf{R}_1\rangle\pi_1 = \langle \mathtt{tt}, \mathsf{R}_m\rangle$ by induction hypothesis (14.1). Then $\mathsf{let}\, m = \min(n, \ell)\ \mathsf{in}\ \forall i \in [1, m]\ .\ \langle \underline{\varrho}, \langle \ell_i, \rho_i\rangle\rangle \in \mathscr{S}^r[\![\mathsf{L}_i : \mathsf{B}_i]\!]$. Moreover, (18) implies that $\mathscr{M}^t\langle \underline{\varrho}, \mathsf{R}\rangle\pi = \langle \mathtt{tt}, \mathsf{R}_m\rangle$

– Otherwise, the specification R_1 is not satisfied from 2 up to m, so, by induction hypothesis (14.2), $\exists i \in [2,m]\ .\ \forall j \in [2,i[\ .\ \langle \underline{\varrho},\ \langle \ell_j,\ \rho_j\rangle\rangle \in \mathcal{S}^r[\![\mathsf{L}_j : \mathsf{B}_j]\!] \wedge \langle \underline{\varrho},\ \langle \ell_i,\ \rho_i\rangle\rangle \notin \mathcal{S}^r[\![\mathsf{L}_i : \mathsf{B}_i]\!]$ and so $\mathcal{M}^t\langle \underline{\varrho},\ \mathsf{R}\rangle\pi_1 = \langle \mathsf{ff},\ \mathsf{R}_i\rangle$. It follows that $\exists i \in [1,\min(n,\ell)]\ .\ \forall j \in [1,i[\ .\ \langle \underline{\varrho},\ \langle \ell_j,\ \rho_j\rangle\rangle \in \mathcal{S}^r[\![\mathsf{L}_j : \mathsf{B}_j]\!] \wedge \langle \underline{\varrho},\ \langle \ell_i,\ \rho_i\rangle\rangle \notin \mathcal{S}^r[\![\mathsf{L}_i : \mathsf{B}_i]\!]$. Moreover, by (18), $\mathcal{M}^t\langle \underline{\varrho},\ \mathsf{R}\rangle\pi = \langle \mathsf{ff},\ \mathsf{R}_i\rangle$.

Let us now prove that $\mathcal{M}[\![\mathsf{P}]\!]\mathsf{R} \subseteq \mathsf{prefix}(\mathcal{S}^r[\![\mathsf{R} \bullet (\mathsf{?} : \mathsf{tt})^*]\!])$. By (22), $\mathsf{dnf}(\mathsf{R}) = (\mathsf{R}_1 \mid \ldots \mid \mathsf{R}_p)$ and, by def. $\cup$, $\exists a \in [1,p]\ .\ \pi \in \mathcal{M}[\![\mathsf{P}]\!]\mathsf{R}_a$ where $\mathsf{R}_a \in \mathcal{R}^{\natural}$ is $\mid$-free by (16). So, by (19), $\exists \mathsf{R}'_a \in \mathcal{R}\ .\ \pi = \langle \ell_1,\ \rho_1\rangle\pi' \in \mathcal{S}^*[\![\mathsf{P}]\!] \wedge \langle \mathsf{tt},\ \mathsf{R}'_a\rangle = \mathcal{M}^t\langle \rho_1,\ \mathsf{R}_a\rangle(\langle \ell_1,\ \rho_1\rangle\pi')$. So we are in case (14.1), and there are two subcases.

- If $n \leqslant \ell$ where $\ell \in \mathbb{N}$, and then $\mathsf{R}_a \eqsim \mathsf{L}_1 : \mathsf{B}_1 \bullet \mathsf{L}_2 : \mathsf{B}_2 \bullet \ldots \bullet \mathsf{L}_n : \mathsf{B}_n \bullet \mathsf{R}'_a$ and $\forall i \in [1,n]\ .\ \langle \rho_1,\ \langle \ell_i,\ \rho_i\rangle\rangle \in \mathcal{S}^r[\![\mathsf{L}_i : \mathsf{B}_i]\!]$. Moreover, by (15), $\forall i \in [n+1,\ell]\ .\ \langle \rho_1,\ \langle \ell_i,\ \rho_i\rangle\rangle \in \mathcal{S}^r[\![\mathsf{?} : \mathsf{tt}]\!]$. Therefore

$$\begin{aligned}
&\langle \rho_1,\ \pi\rangle \in \mathcal{S}^r[\![\mathsf{L}_1 : \mathsf{B}_1]\!] \circledcirc \cdots \circledcirc \mathcal{S}^r[\![\mathsf{L}_n : \mathsf{B}_n]\!] \circledcirc \overbrace{\mathcal{S}^r[\![\mathsf{?} : \mathsf{tt}]\!] \circledcirc \cdots \circledcirc \mathcal{S}^r[\![\mathsf{?} : \mathsf{tt}]\!]}^{n-\ell \text{ times}} \\
&\Rightarrow \langle \rho_1,\ \pi\rangle \in \mathcal{S}^r[\![\mathsf{R}_a \bullet (\mathsf{?} : \mathsf{tt})^{\ell-n}]\!] && \text{(15) for } \bullet \\
&\Rightarrow \langle \rho_1,\ \pi\rangle \in \mathcal{S}^r[\![\mathsf{R} \bullet (\mathsf{?} : \mathsf{tt})^{\ell-n}]\!] && \text{(15) for } \mid \\
&\Rightarrow \langle \rho_1,\ \pi\rangle \in \mathsf{prefix}(\mathcal{S}^r[\![\mathsf{R} \bullet (\mathsf{?} : \mathsf{tt})^{\ell-n}]\!]) && \text{prefix is } \subseteq\text{-extensive} \\
&\Rightarrow \langle \rho_1,\ \pi\rangle \in \mathsf{prefix}(\mathcal{S}^r[\![\mathsf{R} \bullet (\mathsf{?} : \mathsf{tt})^*]\!]) && \text{(15) for } {}^* \text{ and prefix is } \subseteq\text{-increasing}
\end{aligned}$$

- Otherwise, $n > \ell$, and then $\mathsf{R}_a \eqsim \mathsf{L}_1 : \mathsf{B}_1 \bullet \mathsf{L}_2 : \mathsf{B}_2 \bullet \ldots \bullet \mathsf{L}_\ell : \mathsf{B}_\ell \bullet \mathsf{R}'_a$ and $\forall i \in [1,\ell]\ .\ \langle \rho_1,\ \langle \ell_i,\ \rho_i\rangle\rangle \in \mathcal{S}^r[\![\mathsf{L}_i : \mathsf{B}_i]\!]$. Therefore

$$\begin{aligned}
&\langle \rho_1,\ \pi\rangle \in \mathcal{S}^r[\![\mathsf{L}_1 : \mathsf{B}_1]\!] \circledcirc \cdots \circledcirc \mathcal{S}^r[\![\mathsf{L}_\ell : \mathsf{B}_\ell]\!] \\
&\Rightarrow \langle \rho_1,\ \pi\rangle \in \mathcal{S}^r[\![\mathsf{L}_1 : \mathsf{B}_1 \bullet \cdots \bullet \mathsf{L}_\ell : \mathsf{B}_\ell]\!] && \text{(15) for } \bullet \\
&\Rightarrow \langle \rho_1,\ \pi\rangle \in \mathsf{prefix}(\mathcal{S}^r[\![\mathsf{R} \bullet (\mathsf{?} : \mathsf{tt})^*]\!]) && \text{(15) for } \bullet \text{ and } {}^* \text{ and def. prefix}
\end{aligned}$$ □

Lemma 15 (Model checking completeness). $(\{\underline{\varrho}\} \times \mathcal{S}^*[\![\mathsf{P}]\!]) \subseteq \mathsf{prefix}(\mathcal{S}^r[\![\mathsf{R} \bullet (\mathsf{?} : \mathsf{tt})^*]\!]) \Rightarrow (\mathcal{S}^*[\![\mathsf{P}]\!] \subseteq \mathcal{M}[\![\mathsf{P}]\!]\mathsf{R})$.

Proof (of Lem. 15). By contradiction, assume that $\pi = \langle \ell_1, \rho_1\rangle\pi' \in \mathcal{S}^*[\![\mathsf{P}]\!]$ and $\langle \underline{\varrho},\ \pi\rangle \notin \mathsf{prefix}(\mathcal{S}^r[\![\mathsf{R} \bullet (\mathsf{?} : \mathsf{tt})^*]\!])$. By decomposition of R by fstnxt as in the proof of Lem. 14, we have $\exists i \in [1,\min(n,\ell)]\ .\ \forall j \in [1,i[\ .\ \langle \underline{\varrho},\ \langle \ell_j,\ \rho_j\rangle\rangle \in \mathcal{S}^r[\![\mathsf{L}_j : \mathsf{B}_j]\!] \wedge \langle \underline{\varrho},\ \langle \ell_i,\ \rho_i\rangle\rangle$. Therefore, we are in case (14.2) and so $\mathcal{M}^t\langle \underline{\varrho},\ \mathsf{R}\rangle\pi = \langle \mathsf{ff},\ \mathsf{R}_i\rangle$. By (19), $\pi \notin \mathcal{M}[\![\mathsf{P}]\!]\mathsf{R}$.

Proof (of Th. 13). Follows from Lem. 14 and Lem. 15.

$$\begin{aligned}
&\mathsf{P}, \underline{\varrho} \vDash \mathsf{R} \\
&\Leftrightarrow (\{\underline{\varrho}\} \times \mathcal{S}^*[\![\mathsf{P}]\!]) \subseteq \mathsf{prefix}(\mathcal{S}^r[\![\mathsf{R} \bullet (\mathsf{?} : \mathsf{tt})^*]\!]) && \text{Def. 7} \\
&\Leftrightarrow (\{\underline{\varrho}\} \times \mathcal{S}^*[\![\mathsf{P}]\!]) \subseteq \mathcal{M}\,\langle \underline{\varrho},\ \mathsf{R}\rangle(\mathcal{S}^*[\![\mathsf{P}]\!]) && \text{Lem. 14 for } \Leftarrow \text{ and Lem. 15 for } \Rightarrow \\
&\Leftrightarrow \alpha_{\mathcal{M}\,\langle \underline{\varrho},\ \mathsf{R}\rangle}(\mathcal{S}^*[\![\mathsf{P}]\!]) && \text{def. (25) of } \alpha_{\mathcal{M}\,\langle \underline{\varrho},\ \mathsf{R}\rangle}(X) \triangleq (\{\underline{\varrho}\} \times X) \subseteq \mathcal{M}\,\langle \underline{\varrho},\ \mathsf{R}\rangle(X)
\end{aligned}$$ □

At this point we know, by (22) and Th. 13 that a model checker $\mathscr{M}[\![\mathsf{S}]\!]\langle \underline{\varrho}, \mathsf{R}\rangle$ is a sound and complete abstraction $\mathscr{M}\langle \underline{\varrho}, \mathsf{R}\rangle(\mathscr{S}^*[\![\mathsf{S}]\!])$ of the program component semantics $\mathscr{S}^*[\![\mathsf{S}]\!]$ which provides a counter example in case of failure. This allows us to derive a structural model checker $\widehat{\mathscr{M}}[\![\mathsf{P}]\!]\langle \underline{\varrho}, \mathsf{R}\rangle$ in Section **8** by calculational design.

8. Structural model checking

By Def. 7 of the model checking of $\mathsf{S}, \underline{\varrho} \vDash \mathsf{R}$ of a program $\mathsf{P} \in \mathbb{P}$ for a regular specification $\mathsf{R} \in \mathbb{R}^+$ and initial environment $\underline{\varrho}$, Th. 13 shows that a model checker $\mathscr{M}[\![\mathsf{P}]\!]\langle \underline{\varrho}, \mathsf{R}\rangle$ is a sound and complete abstraction $\mathscr{M}\langle \underline{\varrho}, \mathsf{R}\rangle(\mathscr{S}^*[\![\mathsf{P}]\!])$ of the program semantics $\mathscr{S}^*[\![\mathsf{P}]\!]$. This abstraction does not provide a model checking algorithm specification.

The standard model checking algorithms [10] use a transition system (or a Kripke structure variation [35]) for hardware and software modeling and proceed by induction on computation steps.

In contrast, we proceed by structural induction on programs, which will be shown in Th. 16 to be logically equivalent (but maybe more efficient since fixpoints are computed locally). The structural model checking $\widehat{\mathscr{M}}[\![\mathsf{P}]\!]\langle \underline{\varrho}, \mathsf{R}\rangle$ of the program P proceeds by structural induction on the program structure:

$$\begin{cases} \widehat{\mathscr{M}}[\![\mathsf{S}]\!]\langle \underline{\varrho}, \mathsf{R}\rangle & \triangleq \quad \widehat{\mathscr{F}}[\![\mathsf{S}]\!](\prod\limits_{\mathsf{S}' \lhd \mathsf{S}} \widehat{\mathscr{M}}[\![\mathsf{S}']\!])\langle \underline{\varrho}, \mathsf{R}\rangle \\ \mathsf{S} \in \mathbb{P}\mathfrak{c} \end{cases}$$

where the transformer $\widehat{\mathscr{F}}$ uses the results of model checking of the immediate components $\mathsf{S}' \lhd \mathsf{S}$ and involves a fixpoint computation for iteration statements.

The following Th. 16 shows that the algorithm specification is correct, that is $\widehat{\mathscr{M}}[\![\mathsf{S}]\!] = \mathscr{M}[\![\mathsf{S}]\!]$ for all program components S. So together with Th. 13, the structural model checking is proved sound and complete.

Theorem 16. $\forall \mathsf{S} \in \mathbb{P}\mathfrak{c}, \mathsf{R} \in \mathbb{R}, \underline{\varrho} \in \mathbb{E}^\forall . \widehat{\mathscr{M}}^\downarrow[\![\mathsf{S}]\!]\langle \underline{\varrho}, \mathsf{R}\rangle = \mathscr{M}^\downarrow[\![\mathsf{S}]\!]\langle \underline{\varrho}, \mathsf{R}\rangle$ and $\widehat{\mathscr{M}}[\![\mathsf{S}]\!]\langle \underline{\varrho}, \mathsf{R}\rangle = \mathscr{M}[\![\mathsf{S}]\!]\langle \underline{\varrho}, \mathsf{R}\rangle$.

The proof of Th. 16 is by calculational design and proceeds by structural induction on the program component S. Assuming $\mathscr{M}[\![\mathsf{S}']\!] = \widehat{\mathscr{M}}[\![\mathsf{S}']\!]$ for all immediate components $\mathsf{S}' \lhd \mathsf{S}$ of statement, the proof for each program component S has the following form.

$$\begin{array}{ll} & \mathscr{M}[\![\mathsf{S}]\!]\langle \underline{\varrho}, \mathsf{R}\rangle \\ \triangleq & \mathscr{M}\langle \underline{\varrho}, \mathsf{R}\rangle(\mathscr{S}^*[\![\mathsf{S}]\!]) \qquad \wr(22)\wr \\ = & \mathscr{M}\langle \underline{\varrho}, \mathsf{R}\rangle(\mathscr{F}[\![\mathsf{S}]\!](\prod\limits_{\mathsf{S}' \lhd \mathsf{S}} \mathscr{S}^*[\![\mathsf{S}']\!])\langle \underline{\varrho}, \mathsf{R}\rangle) \\ & \quad \wr \text{by structural definition } \mathscr{S}^*[\![\mathsf{S}]\!] = \mathscr{F}[\![\mathsf{S}]\!](\textstyle\prod_{\mathsf{S}' \lhd \mathsf{S}} \mathscr{S}^*[\![\mathsf{S}']\!]) \text{ of the prefix trace semantics in Section } \mathbf{2}\wr \end{array}$$

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

$= \widehat{\mathscr{F}}[\![\mathsf{S}]\!](\prod_{\mathsf{S}' \lhd \mathsf{S}} \mathscr{M}[\![\mathsf{S}']\!])\langle \underline{\varrho}, \mathsf{R}\rangle$

⟅by calculational design to commute the model checking abstraction on the result to the model checking of the arguments of $\mathscr{S}^*[\![\mathsf{S}]\!]$⟆

$= \widehat{\mathscr{F}}[\![\mathsf{S}]\!](\prod_{\mathsf{S}' \lhd \mathsf{S}} \widehat{\mathscr{M}}[\![\mathsf{S}']\!])\langle \underline{\varrho}, \mathsf{R}\rangle$ ⟅ind. hyp.⟆

$\triangleq \widehat{\mathscr{M}}[\![\mathsf{S}]\!]\langle \underline{\varrho}, \mathsf{R}\rangle$ ⟅by defining $\widehat{\mathscr{M}}[\![\mathsf{S}]\!] \triangleq \widehat{\mathscr{F}}[\![\mathsf{S}]\!](\prod_{\mathsf{S}' \lhd \mathsf{S}} \widehat{\mathscr{M}}[\![\mathsf{S}']\!])$⟆

For iteration statements, $\mathscr{F}[\![\mathsf{S}]\!](\prod_{\mathsf{S}' \lhd \mathsf{S}} \mathscr{S}^*[\![\mathsf{S}']\!])\langle \underline{\varrho}, \mathsf{R}\rangle$ is a fixpoint, and this proof involves the fixpoint transfer theorem [16, Th. 7.1.0.4 (3)] based on the commutation of the concrete and abstract transformer with the abstraction. The calculational design of the structural model checking $\widehat{\mathscr{M}}[\![\mathsf{S}]\!]$ is shown below.

Definition 17 (Structural model checking).

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

$$\widehat{\mathscr{M}}[\![\mathsf{P}]\!]\langle \underline{\varrho}, \mathsf{R}\rangle \triangleq \mathsf{let}\,(\mathsf{R}_1 \mid \ldots \mid \mathsf{R}_n) = \mathsf{dnf}(\mathsf{R})\ \mathsf{in} \bigcup_{i=1}^{n} \{\pi \mid \exists \mathsf{R}' \in \mathbb{R} \,.\, \langle \pi, \mathsf{R}'\rangle \in \widehat{\mathscr{M}}^{\downarrow}[\![\mathtt{Sl}]\!]\langle \underline{\varrho}, \mathsf{R}_i\rangle\} \tag{26}$$

Proof. — In case (26) of a program $\mathsf{P} ::= \mathtt{Sl}\ \ell$, the calculational design is as follows.

$\mathscr{M}[\![\mathsf{P}]\!]\langle \underline{\varrho}, \mathsf{R}\rangle$

$\triangleq \mathscr{M}\langle \underline{\varrho}, \mathsf{R}\rangle(\mathscr{S}^*[\![\mathsf{P}]\!])$ ⟅(22)⟆

$= \mathsf{let}\,(\mathsf{R}_1 \mid \ldots \mid \mathsf{R}_n) = \mathsf{dnf}(\mathsf{R})\ \mathsf{in} \bigcup_{i=1}^{n} \{\pi \mid \exists \mathsf{R}' \in \mathbb{R} \,.\, \langle \pi, \mathsf{R}'\rangle \in \mathscr{M}^{\downarrow}\langle \underline{\varrho}, \mathsf{R}_i\rangle(\mathscr{S}^*[\![\mathsf{P}]\!])\}$ ⟅(21)⟆

$= \mathsf{let}\,(\mathsf{R}_1 \mid \ldots \mid \mathsf{R}_n) = \mathsf{dnf}(\mathsf{R})\ \mathsf{in} \bigcup_{i=1}^{n} \{\pi \mid \exists \mathsf{R}' \in \mathbb{R} \,.\, \langle \pi, \mathsf{R}'\rangle \in \mathscr{M}^{\downarrow}\langle \underline{\varrho}, \mathsf{R}_i\rangle(\mathscr{S}^*[\![\mathtt{Sl}]\!])\}$ ⟅def. (9) of $\mathscr{S}^*[\![\mathsf{P}]\!] \triangleq \mathscr{S}^*[\![\mathtt{Sl}]\!]$⟆

$= \mathsf{let}\,(\mathsf{R}_1 \mid \ldots \mid \mathsf{R}_n) = \mathsf{dnf}(\mathsf{R})\ \mathsf{in} \bigcup_{i=1}^{n} \{\pi \mid \exists \mathsf{R}' \in \mathbb{R} \,.\, \langle \pi, \mathsf{R}'\rangle \in \widehat{\mathscr{M}}^{\downarrow}\langle \underline{\varrho}, \mathsf{R}_i\rangle(\mathscr{S}^*[\![\mathtt{Sl}]\!])\}$ ⟅ind. hyp.⟆

$= \mathsf{let}\,(\mathsf{R}_1 \mid \ldots \mid \mathsf{R}_n) = \mathsf{dnf}(\mathsf{R})\ \mathsf{in} \bigcup_{i=1}^{n} \{\pi \mid \exists \mathsf{R}' \in \mathbb{R} \,.\, \langle \pi, \mathsf{R}'\rangle \in \widehat{\mathscr{M}}^{\downarrow}[\![\mathtt{Sl}]\!]\langle \underline{\varrho}, \mathsf{R}_i\rangle\}$ ⟅(20)⟆

$= \widehat{\mathscr{M}}[\![\mathsf{P}]\!]\langle \underline{\varrho}, \mathsf{R}\rangle$ ⟅(26)⟆ □

Definition 17 (Structural model checking, contn'd)

- Model checking an empty temporal specification ε.

$$\widehat{\mathscr{M}}^{\downarrow}[\![\mathtt{S}]\!]\langle \underline{\varrho},\, \varepsilon\rangle \quad \triangleq \quad \{\langle \pi,\, \varepsilon\rangle \mid \pi \in \mathscr{S}^{*}[\![\mathtt{S}]\!]\} \tag{27}$$

- It is assumed below that $\mathtt{R} \in \mathscr{R}^{\downarrow} \cap \mathscr{R}^{+}$ is a non-empty, alternative $\mathtt{|}$-free regular expression.

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

$$\begin{aligned}\widehat{\mathscr{M}}^{\downarrow}[\![\mathtt{Sl}]\!]\langle \underline{\varrho},\, \mathtt{R}\rangle \quad &\triangleq \quad \widehat{\mathscr{M}}^{\downarrow}[\![\mathtt{Sl}']\!]\langle \underline{\varrho},\, \mathtt{R}\rangle \\ &\cup \{\langle \pi \cdot \langle \mathsf{at}[\![\mathtt{S}]\!],\, \rho\rangle \cdot \pi',\, \mathtt{R}''\rangle \mid \langle \pi \cdot \langle \mathsf{at}[\![\mathtt{S}]\!],\, \rho\rangle,\, \mathtt{R}'\rangle \in \widehat{\mathscr{M}}^{\downarrow}[\![\mathtt{Sl}']\!]\langle \underline{\varrho},\, \mathtt{R}\rangle \wedge \\ &\qquad \langle \langle \mathsf{at}[\![\mathtt{S}]\!],\, \rho\rangle \cdot \pi',\, \mathtt{R}''\rangle \in \widehat{\mathscr{M}}^{\downarrow}[\![\mathtt{S}]\!]\langle \underline{\varrho},\, \mathtt{R}'\rangle\}\end{aligned} \tag{28}$$

- Model checking an empty statement list $\mathtt{Sl} ::= \epsilon$

$$\begin{aligned}\widehat{\mathscr{M}}^{\downarrow}[\![\mathtt{Sl}]\!]\langle \underline{\varrho},\, \mathtt{R}\rangle \quad &\triangleq \quad \mathsf{let}\ \langle \mathtt{L}\ \mathtt{:}\ \mathtt{B},\, \mathtt{R}'\rangle = \mathsf{fstnxt}(\mathtt{R})\ \mathsf{in} \\ &\qquad \{\langle \langle \mathsf{at}[\![\mathtt{Sl}]\!],\, \rho\rangle,\, \mathtt{R}'\rangle \mid \langle \underline{\varrho},\, \langle \mathsf{at}[\![\mathtt{Sl}]\!],\, \rho\rangle\rangle \in \mathscr{S}^{\mathsf{r}}[\![\mathtt{L}\ \mathtt{:}\ \mathtt{B}]\!]\}\end{aligned} \tag{29}$$

(In practice the empty statement list ϵ needs not be specified so we could eliminate that need by ignoring ϵ in the specification $\mathtt{R}$ and defining $\widehat{\mathscr{M}}^{\downarrow}[\![\mathtt{Sl}]\!]\langle \underline{\varrho},\, \mathtt{R}\rangle \triangleq \{\langle \langle \mathsf{at}[\![\mathtt{Sl}]\!],\, \rho\rangle,\, \mathtt{R}\rangle \mid \rho \in \mathbb{E}\mathsf{v}\}$.)

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

$$\begin{aligned}\widehat{\mathscr{M}}^{\downarrow}[\![\mathtt{S}]\!]\langle \underline{\varrho},\, \mathtt{R}\rangle \triangleq\ &\mathsf{let}\ \langle \mathtt{L}\ \mathtt{:}\ \mathtt{B},\, \mathtt{R}'\rangle = \mathsf{fstnxt}(\mathtt{R})\ \mathsf{in} && (30)\\ &\{\langle \langle \mathsf{at}[\![\mathtt{S}]\!],\, \rho\rangle,\, \mathtt{R}'\rangle \mid \langle \underline{\varrho},\, \langle \mathsf{at}[\![\mathtt{S}]\!],\, \rho\rangle\rangle \in \mathscr{S}^{\mathsf{r}}[\![\mathtt{L}\ \mathtt{:}\ \mathtt{B}]\!]\} && (a)\\ &\cup \{\langle \langle \mathsf{at}[\![\mathtt{S}]\!],\, \rho\rangle\langle \mathsf{aft}[\![\mathtt{S}]\!],\, \rho[\mathtt{x} \leftarrow \mathscr{A}[\![\mathtt{A}]\!]\rho]\rangle,\, \varepsilon\rangle \mid \mathtt{R}' \in \mathscr{R}_{\varepsilon} \wedge && (b)\\ &\qquad \langle \underline{\varrho},\, \langle \mathsf{at}[\![\mathtt{S}]\!],\, \rho\rangle\rangle \in \mathscr{S}^{\mathsf{r}}[\![\mathtt{L}\ \mathtt{:}\ \mathtt{B}]\!]\} \\ &\cup \{\langle \langle \mathsf{at}[\![\mathtt{S}]\!],\, \rho\rangle\langle \mathsf{aft}[\![\mathtt{S}]\!],\, \rho[\mathtt{x} \leftarrow \mathscr{A}[\![\mathtt{A}]\!]\rho]\rangle,\, \mathtt{R}''\rangle \mid \mathtt{R}' \notin \mathscr{R}_{\varepsilon} \wedge && (c)\\ &\qquad \langle \underline{\varrho},\, \langle \mathsf{at}[\![\mathtt{S}]\!],\, \rho\rangle\rangle \in \mathscr{S}^{\mathsf{r}}[\![\mathtt{L}\ \mathtt{:}\ \mathtt{B}]\!] \wedge \langle \mathtt{L}'\ \mathtt{:}\ \mathtt{B}',\, \mathtt{R}''\rangle = \mathsf{fstnxt}(\mathtt{R}') \wedge \\ &\qquad\quad \langle \underline{\varrho},\, \langle \mathsf{aft}[\![\mathtt{S}]\!],\, \rho[\mathtt{x} \leftarrow \mathscr{A}[\![\mathtt{A}]\!]\rho]\rangle\rangle \in \mathscr{S}^{\mathsf{r}}[\![\mathtt{L}'\ \mathtt{:}\ \mathtt{B}']\!]\}\end{aligned}$$

For the assignment $\mathtt{S} ::= {}^{\ell}\ \mathtt{x = A\ ;}$ in (30), case (a) checks the prefixes that stops at ℓ whereas (b) and (c) checks the maximal traces stopping after the assignment. In each trace checked for the specification $\mathtt{R}$, the states are checked successively and the continuation specification is returned together with the checked trace, unless the check fails. Checking the assignment $\mathtt{S} ::= {}^{\ell}\ \mathtt{x = A\ ;}$ in (30) for $\langle \mathtt{L}\ \mathtt{:}\ \mathtt{B},\, \mathtt{R}'\rangle = \mathsf{fstnxt}(\mathtt{R})$ consists in first checking $\mathtt{L}\ \mathtt{:}\ \mathtt{B}$ at ℓ and then

checking on $\mathtt{R}'$ after the statement. In case (b), $\mathtt{R}'$ is empty so trivially satisfied. Otherwise, in case (c), $\langle \mathtt{L}' : \mathtt{B}', \mathtt{R}'' \rangle = \mathsf{fstnxt}(\mathtt{R}')$ so $\mathtt{L}' : \mathtt{B}'$ is checked after the statement while $\mathtt{R}''$ is the continuation specification.

PROOF. — In case (30) of an assignment statement $\mathtt{S} ::= \ell\ \mathtt{x = A\ ;}$, the calculational design is as follows.

$$
\begin{aligned}
& \mathscr{M}^{\natural}[\![\mathtt{S}]\!]\ \langle \underline{\varrho},\ \mathtt{R} \rangle \\
= {} & \{\langle \pi,\ \mathtt{R}' \rangle \mid \pi \in \mathscr{S}^{*}[\![\mathtt{Sl}]\!] \wedge \langle \mathtt{tt},\ \mathtt{R}' \rangle = \mathscr{M}^{t}\langle \underline{\varrho},\ \mathtt{R} \rangle \pi\} && \text{(20) and (19)} \\
= {} & \{\langle \pi,\ \mathtt{R}' \rangle \mid \pi \in \{\langle \ell,\ \rho \rangle \mid \rho \in \mathbb{E}\mathtt{v}\} \cup \{\langle \ell,\ \rho \rangle \langle \mathsf{aft}[\![\mathtt{S}]\!],\ \rho[\mathtt{x} \leftarrow v] \rangle \mid \rho \in \mathbb{E}\mathtt{v} \wedge v = \\
& \mathscr{A}[\![\mathtt{A}]\!]\rho \wedge \langle \mathtt{tt},\ \mathtt{R}' \rangle = \mathscr{M}^{t}\langle \underline{\varrho},\ \mathtt{R} \rangle \pi\} && \text{(1)} \\
= {} & \{\langle \langle \ell,\ \rho \rangle,\ \mathtt{R}' \rangle \mid \rho \in \mathbb{E}\mathtt{v} \wedge \langle \mathtt{tt},\ \mathtt{R}' \rangle = \mathscr{M}^{t}\langle \underline{\varrho},\ \mathtt{R} \rangle \langle \ell,\ \rho \rangle\} \cup \\
& \{\langle \langle \ell,\ \rho \rangle \langle \mathsf{aft}[\![\mathtt{S}]\!],\ \rho[\mathtt{x} \leftarrow v] \rangle,\ \mathtt{R}' \rangle \mid \rho \in \mathbb{E}\mathtt{v} \wedge v = \mathscr{A}[\![\mathtt{A}]\!]\rho \langle \mathtt{tt},\ \mathtt{R}' \rangle = \mathscr{M}^{t}\langle \underline{\varrho},\ \mathtt{R} \rangle \langle \ell,\ \\
& \rho \rangle \langle \mathsf{aft}[\![\mathtt{S}]\!],\ \rho[\mathtt{x} \leftarrow v] \rangle\} && \text{def. } \cup \text{ and } \in \\
= {} & \{\langle \langle \ell,\ \rho \rangle,\ \mathtt{R}' \rangle \mid \langle \mathtt{tt},\ \mathtt{R}' \rangle = \mathsf{let}\ \langle \mathtt{L} : \mathtt{B},\ \mathtt{R}'' \rangle = \mathsf{fstnxt}(\mathtt{R})\ \mathsf{in}\ (\!|\ \langle \underline{\varrho},\ \langle \ell,\ \rho \rangle \rangle \in \mathscr{S}^{r}[\![\mathtt{L} : \mathtt{B}]\!]\ ?\ \\
& \langle \mathtt{tt},\ \mathtt{R}'' \rangle : \langle \mathtt{ff},\ \mathtt{R}' \rangle\ |\!)\} \cup \\
& \{\langle \langle \ell,\ \rho \rangle \langle \mathsf{aft}[\![\mathtt{S}]\!],\ \rho[\mathtt{x} \leftarrow v] \rangle,\ \mathtt{R}' \rangle \mid v = \mathscr{A}[\![\mathtt{A}]\!]\rho \wedge \langle \mathtt{tt},\ \mathtt{R}' \rangle = \mathsf{let}\ \langle \mathtt{L} : \mathtt{B},\ \mathtt{R}'' \rangle = \\
& \mathsf{fstnxt}(\mathtt{R})\ \mathsf{in}\ (\!|\ \langle \underline{\varrho},\ \langle \ell,\ \rho \rangle \rangle \in \mathscr{S}^{r}[\![\mathtt{L} : \mathtt{B}]\!]\ ?\ \mathscr{M}^{t}\langle \underline{\varrho},\ \mathtt{R}'' \rangle \langle \mathsf{aft}[\![\mathtt{S}]\!],\ \rho[\mathtt{x} \leftarrow v] \rangle : \langle \mathtt{ff},\ \mathtt{R}'' \rangle\ |\!)\} \\
& && \text{(18)} \\
= {} & \{\langle \langle \ell,\ \rho \rangle,\ \mathtt{R}' \rangle \mid \langle \mathtt{L} : \mathtt{B},\ \mathtt{R}' \rangle = \mathsf{fstnxt}(\mathtt{R}) \wedge \langle \underline{\varrho},\ \langle \ell,\ \rho \rangle \rangle \in \mathscr{S}^{r}[\![\mathtt{L} : \mathtt{B}]\!]\} \cup \\
& \{\langle \langle \ell,\ \rho \rangle \langle \mathsf{aft}[\![\mathtt{S}]\!],\ \rho[\mathtt{x} \leftarrow v] \rangle,\ \mathtt{R}' \rangle \mid v = \mathscr{A}[\![\mathtt{A}]\!]\rho \wedge \exists \mathtt{R}'' \in \mathscr{R}\ .\ \langle \mathtt{L} : \mathtt{B},\ \mathtt{R}'' \rangle = \mathsf{fstnxt}(\mathtt{R}) \wedge \\
& \langle \underline{\varrho},\ \langle \ell,\ \rho \rangle \rangle \in \mathscr{S}^{r}[\![\mathtt{L} : \mathtt{B}]\!] \wedge (\!|\ \mathtt{R}'' \in \mathscr{R}_{\varepsilon}\ ?\ \mathtt{tt} : \mathscr{M}^{t}\langle \underline{\varrho},\ \mathtt{R}'' \rangle \langle \mathsf{aft}[\![\mathtt{S}]\!],\ \rho[\mathtt{x} \leftarrow v] \rangle = \langle \mathtt{tt},\ \\
& \mathtt{R}' \rangle\ |\!)\} && \text{def. = and } \mathscr{M}^{t}\langle \underline{\varrho},\ \varepsilon \rangle \pi \triangleq \langle \mathtt{tt},\ \varepsilon \rangle \text{ by (18)} \\
= {} & \{\langle \langle \ell,\ \rho \rangle,\ \mathtt{R}' \rangle \mid \langle \mathtt{L} : \mathtt{B},\ \mathtt{R}' \rangle = \mathsf{fstnxt}(\mathtt{R}) \wedge \langle \underline{\varrho},\ \langle \ell,\ \rho \rangle \rangle \in \mathscr{S}^{r}[\![\mathtt{L} : \mathtt{B}]\!]\} \cup \\
& \{\langle \langle \ell,\ \rho \rangle \langle \mathsf{aft}[\![\mathtt{S}]\!],\ \rho[\mathtt{x} \leftarrow v] \rangle,\ \mathtt{R}' \rangle \mid v = \mathscr{A}[\![\mathtt{A}]\!]\rho \wedge \exists \mathtt{R}'' \in \mathscr{R}\ .\ \langle \mathtt{L} : \mathtt{B},\ \mathtt{R}'' \rangle = \mathsf{fstnxt}(\mathtt{R}) \wedge \\
& \langle \underline{\varrho},\ \langle \ell,\ \rho \rangle \rangle \in \mathscr{S}^{r}[\![\mathtt{L} : \mathtt{B}]\!] \wedge (\!|\ \mathtt{R}'' \in \mathscr{R}_{\varepsilon}\ ?\ \mathtt{tt} : \mathsf{let}\ \langle \mathtt{L}' : \mathtt{B}',\ \mathtt{R}''' \rangle = \mathsf{fstnxt}(\mathtt{R}'')\ \mathsf{in}\ \langle \underline{\varrho},\ \\
& \langle \mathsf{aft}[\![\mathtt{S}]\!],\ \rho[\mathtt{x} \leftarrow v] \rangle \rangle \in \mathscr{S}^{r}[\![\mathtt{L}' : \mathtt{B}']\!]\ |\!)\} && \text{(18)} \\
= {} & \mathsf{let}\ \langle \mathtt{L} : \mathtt{B},\ \mathtt{R}' \rangle = \mathsf{fstnxt}(\mathtt{R})\ \mathsf{in} \\
& \{\langle \langle \ell,\ \rho \rangle,\ \mathtt{R}' \rangle \mid \langle \underline{\varrho},\ \langle \ell,\ \rho \rangle \rangle \in \mathscr{S}^{r}[\![\mathtt{L} : \mathtt{B}]\!]\} \\
& \cup \{\langle \langle \ell,\ \rho \rangle \langle \mathsf{aft}[\![\mathtt{S}]\!],\ \rho[\mathtt{x} \leftarrow v] \rangle,\ \varepsilon \rangle \mid v = \mathscr{A}[\![\mathtt{A}]\!]\rho \wedge \langle \underline{\varrho},\ \langle \ell,\ \rho \rangle \rangle \in \mathscr{S}^{r}[\![\mathtt{L} : \mathtt{B}]\!] \wedge \mathtt{R}' \in \mathscr{R}_{\varepsilon}\} \\
& \cup \{\langle \langle \ell,\ \rho \rangle \langle \mathsf{aft}[\![\mathtt{S}]\!],\ \rho[\mathtt{x} \leftarrow v] \rangle,\ \mathtt{R}'' \rangle \mid v = \mathscr{A}[\![\mathtt{A}]\!]\rho \wedge \langle \underline{\varrho},\ \langle \ell,\ \rho \rangle \rangle \in \mathscr{S}^{r}[\![\mathtt{L} : \mathtt{B}]\!] \wedge \mathtt{R}' \notin \\
& \mathscr{R}_{\varepsilon} \wedge \mathsf{let}\ \langle \mathtt{L}' : \mathtt{B}',\ \mathtt{R}'' \rangle = \mathsf{fstnxt}(\mathtt{R}')\ \mathsf{in}\ \langle \underline{\varrho},\ \langle \mathsf{aft}[\![\mathtt{S}]\!],\ \rho[\mathtt{x} \leftarrow v] \rangle \rangle \in \mathscr{S}^{r}[\![\mathtt{L}' : \mathtt{B}']\!]\} \\
& && \text{def. } \cup \\
= {} & \widehat{\mathscr{M}}^{\natural}[\![\mathtt{S}]\!]\ \langle \underline{\varrho},\ \mathtt{R} \rangle && \text{(30)}\ \square
\end{aligned}
$$

Definition 17 (Structural model checking, continued)

- Model checking a conditional statement $\mathtt{S} ::= \mathtt{if}\ \ell\ \mathtt{(B)\ S}_t$

$$
\begin{aligned}
\widehat{\mathscr{M}}^{\downarrow}[\![\mathsf{S}]\!]\langle \underline{\varrho},\ \mathsf{R}\rangle \triangleq\ & \mathsf{let}\ \langle \mathsf{L}'\ \mathtt{:}\ \mathsf{B}',\ \mathsf{R}'\rangle = \mathsf{fstnxt}(\mathsf{R})\ \mathsf{in} \\
& \quad \{\langle\langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle,\ \mathsf{R}'\rangle \mid \langle \underline{\varrho},\ \langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle\rangle \in \mathscr{S}^{\mathsf{r}}[\![\mathsf{L}'\ \mathtt{:}\ \mathsf{B}']\!]\} \\
& \cup \{\langle\langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle\langle \mathsf{at}[\![\mathsf{S}_t]\!],\ \rho\rangle\pi,\ \mathsf{R}''\rangle \mid \mathscr{B}[\![\mathsf{B}]\!]\rho = \mathtt{tt} \wedge \\
& \qquad \langle \underline{\varrho},\ \langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle\rangle \in \mathscr{S}^{\mathsf{r}}[\![\mathsf{L}'\ \mathtt{:}\ \mathsf{B}']\!] \wedge \\
& \qquad\quad \langle\langle \mathsf{at}[\![\mathsf{S}_t]\!],\ \rho\rangle\pi,\ \mathsf{R}''\rangle \in \widehat{\mathscr{M}}^{\downarrow}[\![\mathsf{S}_t]\!]\langle \underline{\varrho},\ \mathsf{R}'\rangle\} \\
& \cup \{\langle\langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle\langle \mathsf{aft}[\![\mathsf{S}]\!],\ \rho\rangle,\ \varepsilon\rangle \mid\ \mid \mathscr{B}[\![\mathsf{B}]\!]\rho = \mathtt{ff} \wedge \mathsf{R}' \in \mathit{R}_{\varepsilon} \wedge \\
& \qquad \langle \underline{\varrho},\ \langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle\rangle \in \mathscr{S}^{\mathsf{r}}[\![\mathsf{L}'\ \mathtt{:}\ \mathsf{B}']\!]\} \\
& \cup \{\langle\langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle\langle \mathsf{aft}[\![\mathsf{S}]\!],\ \rho\rangle,\ \mathsf{R}''\rangle \mid \mathscr{B}[\![\mathsf{B}]\!]\rho = \mathtt{ff} \wedge \mathsf{R}' \notin \mathit{R}_{\varepsilon} \wedge \\
& \qquad \langle \underline{\varrho},\ \langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle\rangle \in \mathscr{S}^{\mathsf{r}}[\![\mathsf{L}'\ \mathtt{:}\ \mathsf{B}']\!] \wedge \langle \mathsf{L}''\ \mathtt{:}\ \mathsf{B}'',\ \mathsf{R}''\rangle = \mathsf{fstnxt}(\mathsf{R}') \wedge \\
& \qquad\quad \langle \underline{\varrho},\ \langle \mathsf{aft}[\![\mathsf{S}]\!],\ \rho\rangle\rangle \in \mathscr{S}^{\mathsf{r}}[\![\mathsf{L}''\ \mathtt{:}\ \mathsf{B}'']\!]\}
\end{aligned}
\tag{31}
$$

- Model checking a break statement S ::= $^{\ell}$**break ;**

$$
\begin{aligned}
\widehat{\mathscr{M}}^{\downarrow}[\![\mathsf{S}]\!]\langle \underline{\varrho},\ \mathsf{R}\rangle \triangleq\ & \mathsf{let}\ \langle \mathsf{L}\ \mathtt{:}\ \mathsf{B},\ \mathsf{R}'\rangle = \mathsf{fstnxt}(\mathsf{R})\ \mathsf{in} \\
& \quad \{\langle\langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle,\ \mathsf{R}'\rangle \mid \langle \underline{\varrho},\ \langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle\rangle \in \mathscr{S}^{\mathsf{r}}[\![\mathsf{L}\ \mathtt{:}\ \mathsf{B}]\!]\} \\
& \cup \{\langle\langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle\langle \mathsf{brk\text{-}to}[\![\mathsf{S}]\!],\ \rho\rangle,\ \varepsilon\rangle \mid \mathsf{R}' \in \mathit{R}_{\varepsilon} \wedge \\
& \qquad \langle \underline{\varrho},\ \langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle\rangle \in \mathscr{S}^{\mathsf{r}}[\![\mathsf{L}\ \mathtt{:}\ \mathsf{B}]\!]\} \\
& \cup \{\langle\langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle\langle \mathsf{brk\text{-}to}[\![\mathsf{S}]\!],\ \rho\rangle,\ \mathsf{R}''\rangle \mid \mathsf{R}' \notin \mathit{R}_{\varepsilon} \wedge \\
& \qquad \langle \underline{\varrho},\ \langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle\rangle \in \mathscr{S}^{\mathsf{r}}[\![\mathsf{L}\ \mathtt{:}\ \mathsf{B}]\!] \wedge \langle \mathsf{L}'\ \mathtt{:}\ \mathsf{B}',\ \mathsf{R}''\rangle = \mathsf{fstnxt}(\mathsf{R}') \wedge \\
& \qquad\quad \langle \underline{\varrho},\ \langle \mathsf{brk\text{-}to}[\![\mathsf{S}]\!],\ \rho\rangle\rangle \in \mathscr{S}^{\mathsf{r}}[\![\mathsf{L}'\ \mathtt{:}\ \mathsf{B}']\!]\}
\end{aligned}
\tag{32}
$$

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

$$\widehat{\mathscr{M}}^{\natural}[\![\mathsf{S}]\!]\langle \underline{\varrho},\ \mathsf{R}\rangle \ \triangleq\ \mathsf{lfp}^{\sqsubseteq}\,(\widehat{\mathscr{F}}^{\natural}[\![\mathsf{S}]\!]\langle \underline{\varrho},\ \mathsf{R}\rangle) \tag{33}$$

$$\widehat{\mathscr{F}}^{\natural}[\![\mathsf{S}]\!]\langle \underline{\varrho},\ \mathsf{R}\rangle\ X \ \triangleq\ \mathsf{let}\ \langle \mathsf{L}' : \mathsf{B}',\ \mathsf{R}'\rangle = \mathsf{fstnxt}(\mathsf{R})\ \mathsf{in} \tag{34}$$

$$\{\langle\langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle,\ \mathsf{R}'\rangle \mid \rho \in \mathbb{E}\mathsf{v} \wedge \langle \underline{\varrho},\ \langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle\rangle \in \mathscr{S}^{\mathsf{r}}[\![\mathsf{L}' : \mathsf{B}']\!]\} \tag{a}$$

$$\cup\,\{\langle \pi_2\langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle\langle \mathsf{aft}[\![\mathsf{S}]\!],\ \rho\rangle,\ \varepsilon\rangle \mid \langle \pi_2\langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle,\ \varepsilon\rangle \in X \wedge \mathscr{B}[\![\mathsf{B}]\!]\ \rho = \mathsf{ff}\}$$

$$\cup\,\{\langle \pi_2\langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle\langle \mathsf{aft}[\![\mathsf{S}]\!],\ \rho\rangle,\ \varepsilon\rangle \mid \langle \pi_2\langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle,\ \mathsf{R}''\rangle \in X \wedge \mathscr{B}[\![\mathsf{B}]\!]\ \rho = \mathsf{ff} \wedge \mathsf{R}'' \notin \mathscr{R}_\varepsilon \wedge \langle \mathsf{L}' : \mathsf{B}',\ \mathsf{R}'\rangle = \mathsf{fstnxt}(\mathsf{R}'') \wedge \mathsf{R}' \in \mathscr{R}_\varepsilon \wedge \langle \underline{\varrho},\ \langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle\rangle \in \mathscr{S}^{\mathsf{r}}[\![\mathsf{L}' : \mathsf{B}']\!]\} \tag{b}$$

$$\cup\,\{\langle \pi_2\langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle\langle \mathsf{aft}[\![\mathsf{S}]\!],\ \rho\rangle,\ \mathsf{R}'\rangle \mid \langle \pi_2\langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle,\ \mathsf{R}''\rangle \in X \wedge \mathscr{B}[\![\mathsf{B}]\!]\ \rho = \mathsf{ff} \wedge \mathsf{R}'' \notin \mathscr{R}_\varepsilon \wedge \langle \mathsf{L}' : \mathsf{B}',\ \mathsf{R}'''\rangle = \mathsf{fstnxt}(\mathsf{R}'') \wedge \mathsf{R}''' \notin \mathscr{R}_\varepsilon \wedge \langle \underline{\varrho},\ \langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle\rangle \in \mathscr{S}^{\mathsf{r}}[\![\mathsf{L}' : \mathsf{B}']\!] \wedge \langle \mathsf{L}'' : \mathsf{B}'',\ \mathsf{R}'\rangle = \mathsf{fstnxt}(\mathsf{R}''') \wedge \langle \underline{\varrho},\ \langle \mathsf{aft}[\![\mathsf{S}]\!],\ \rho\rangle\rangle \in \mathscr{S}^{\mathsf{r}}[\![\mathsf{L}'' : \mathsf{B}'']\!]\} \tag{c}$$

$$\cup\,\{\langle \pi_2\langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle\langle \mathsf{at}[\![\mathsf{S}_b]\!],\ \rho\rangle\pi_3,\ \varepsilon\rangle \mid \langle \pi_2\langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle,\ \varepsilon\rangle \in X \wedge \mathscr{B}[\![\mathsf{B}]\!]\ \rho = \mathsf{tt} \wedge \langle \mathsf{at}[\![\mathsf{S}_b]\!],\ \rho\rangle\pi_3 \in \mathscr{S}^{*}[\![\mathsf{S}_b]\!]\} \tag{d}$$

$$\cup\,\{\langle \pi_2\langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle\langle \mathsf{at}[\![\mathsf{S}_b]\!],\ \rho\rangle\pi_3,\ \varepsilon\rangle \mid \langle \pi_2\langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle,\ \mathsf{R}''\rangle \in X \wedge \mathscr{B}[\![\mathsf{B}]\!]\ \rho = \mathsf{tt} \wedge \mathsf{R}'' \notin \mathscr{R}_\varepsilon \wedge \langle \mathsf{L} : \mathsf{B},\ \varepsilon\rangle = \mathsf{fstnxt}(\mathsf{R}'') \wedge \langle \underline{\varrho},\ \langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle\rangle \in \mathscr{S}^{\mathsf{r}}[\![\mathsf{L} : \mathsf{B}]\!] \wedge \langle \mathsf{at}[\![\mathsf{S}_b]\!],\ \rho\rangle\pi_3 \in \mathscr{S}^{*}[\![\mathsf{S}_b]\!]\} \tag{e}$$

$$\cup\,\{\langle \pi_2\langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle\langle \mathsf{at}[\![\mathsf{S}_b]\!],\ \rho\rangle\pi_3,\ \mathsf{R}'\rangle \mid \langle \pi_2\langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle,\ \mathsf{R}''\rangle \in X \wedge \mathscr{B}[\![\mathsf{B}]\!]\ \rho = \mathsf{tt} \wedge \mathsf{R}'' \notin \mathscr{R}_\varepsilon \wedge \langle \mathsf{L} : \mathsf{B},\ \mathsf{R}''''\rangle = \mathsf{fstnxt}(\mathsf{R}'') \wedge \langle \underline{\varrho},\ \langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle\rangle \in \mathscr{S}^{\mathsf{r}}[\![\mathsf{L} : \mathsf{B}]\!] \wedge \mathsf{R}'''' \notin \mathscr{R}_\varepsilon \wedge \langle \mathsf{L}' : \mathsf{B}',\ \mathsf{R}'''\rangle = \mathsf{fstnxt}(\mathsf{R}'''') \wedge \langle \underline{\varrho},\ \langle \mathsf{at}[\![\mathsf{S}_b]\!],\ \rho\rangle\rangle \in \mathscr{S}^{\mathsf{r}}[\![\mathsf{L}' : \mathsf{B}']\!] \wedge \langle\langle \mathsf{at}[\![\mathsf{S}_b]\!],\ \rho\rangle\pi_3,\ \mathsf{R}'\rangle \in \mathscr{M}^{\natural}[\![\mathsf{S}_b]\!]\langle \underline{\varrho},\ \mathsf{R}'''\rangle\} \tag{f}$$

— The model checking of an iteration statement $\mathsf{S} ::= \mathtt{while}^{\,\ell}\ \mathtt{(B)}\ \mathsf{S}_b$ in (34) checks one more iteration (after checking the previous ones as recorded by X) while the fixpoint (33) repeats this check for all iterations. Case (a) checks the prefixes that stops at loop entry ℓ. (b) and (c) check the exit of an iteration when the iteration condition is false, (b) when the specification stops at loop entry ℓ before leaving and (c) when the specification goes further. (d), (e) and (f) check one more iteration when the iteration condition is true. In case (d), the continuation after the check of the iterates is empty so trivially satisfied by any continuation of the prefix trace. In case (e), the continuation after the check of the iterates just imposes to verify $\mathsf{L} : \mathsf{B}$ on iteration entry and nothing afterwards. In case (f) the continuation after the check of the iterates requires to verify $\mathsf{L} : \mathsf{B}$ at the loop entry, $\mathsf{L}' : \mathsf{B}'$ at the body entry, and the rest R''' of the specification for the loop body (which returns the possibly empty continuation specification R'). The cases (b) to (f) are mutually exclusive.

The remaining cases are as follows.

Definition 17 (Structural model checking, continued)

- Model checking a skip statement $\mathtt{S} ::= \mathbf{;}$

$$\widehat{\mathscr{M}}^{\downarrow}[\![\mathtt{S}]\!]\langle \underline{\varrho},\, \mathtt{R}\rangle \triangleq \mathsf{let}\ \langle \mathtt{L} : \mathtt{B},\, \mathtt{R}'\rangle = \mathsf{fstnxt}(\mathtt{R})\ \mathsf{in} \quad (35)$$
$$\{\langle\langle \mathsf{at}[\![\mathtt{S}]\!],\, \rho\rangle,\, \mathtt{R}'\rangle \mid \langle \underline{\varrho},\, \langle \mathsf{at}[\![\mathtt{S}]\!],\, \rho\rangle\rangle \in \mathscr{S}^{\mathtt{r}}[\![\mathtt{L} : \mathtt{B}]\!]\}$$

(In practice the skip statement **;** needs not be specified so we could eliminate that need by ignoring **;** in the specification $\mathtt{R}$ and defining $\widehat{\mathscr{M}}^{\downarrow}[\![\mathbf{;}]\!]\langle \underline{\varrho},\, \mathtt{R}\rangle \triangleq \{\langle\langle \mathsf{at}[\![\mathtt{S}]\!],\, \rho\rangle,\, \mathtt{R}\rangle \mid \rho \in \mathbb{E}^{\forall}\}$.)

- Model checking an alternative statement $\mathtt{S} ::= \mathtt{if}^{\,\ell}\ \mathtt{(B)}\ \mathtt{S}_t\ \mathtt{else}\ \mathtt{S}_f$

$$\begin{aligned}\widehat{\mathscr{M}}^{\downarrow}[\![\mathtt{S}]\!]\langle \underline{\varrho},\, \mathtt{R}\rangle \triangleq\ & \mathsf{let}\ \langle \mathtt{L}' : \mathtt{B}',\, \mathtt{R}'\rangle = \mathsf{fstnxt}(\mathtt{R})\ \mathsf{in} \\ & \{\langle\langle \mathsf{at}[\![\mathtt{S}]\!],\, \rho\rangle,\, \mathtt{R}'\rangle \mid \langle \underline{\varrho},\, \langle \mathsf{at}[\![\mathtt{S}]\!],\, \rho\rangle\rangle \in \mathscr{S}^{\mathtt{r}}[\![\mathtt{L}' : \mathtt{B}]\!]\} \\ & \cup \{\langle\langle \mathsf{at}[\![\mathtt{S}]\!],\, \rho\rangle\langle \mathsf{at}[\![\mathtt{S}_t]\!],\, \rho\rangle\pi,\, \mathtt{R}''\rangle \mid \mathscr{B}[\![\mathtt{B}]\!]\rho = \mathtt{tt} \wedge \\ & \quad \langle \underline{\varrho},\, \langle \mathsf{at}[\![\mathtt{S}]\!],\, \rho\rangle\rangle \in \mathscr{S}^{\mathtt{r}}[\![\mathtt{L}' : \mathtt{B}']\!] \wedge \\ & \quad\quad \langle\langle \mathsf{at}[\![\mathtt{S}_t]\!],\, \rho\rangle\pi,\, \mathtt{R}''\rangle \in \widehat{\mathscr{M}}^{\downarrow}[\![\mathtt{S}_t]\!]\langle \underline{\varrho},\, \mathtt{R}'\rangle\} \\ & \cup \{\langle\langle \mathsf{at}[\![\mathtt{S}]\!],\, \rho\rangle\langle \mathsf{at}[\![\mathtt{S}_f]\!],\, \rho\rangle\pi,\, \mathtt{R}''\rangle \mid \mathscr{B}[\![\mathtt{B}]\!]\rho = \mathtt{ff} \wedge \\ & \quad \langle \underline{\varrho},\, \langle \mathsf{at}[\![\mathtt{S}]\!],\, \rho\rangle\rangle \in \mathscr{S}^{\mathtt{r}}[\![\mathtt{L}' : \mathtt{B}']\!] \wedge \\ & \quad\quad \langle\langle \mathsf{at}[\![\mathtt{S}_f]\!],\, \rho\rangle\pi,\, \mathtt{R}''\rangle \in \widehat{\mathscr{M}}^{\downarrow}[\![\mathtt{S}_f]\!]\langle \underline{\varrho},\, \mathtt{R}'\rangle\}\end{aligned} \quad (36)$$

- Model checking a compound statement $\mathtt{S} ::= \mathtt{\{\ Sl\ \}}$

$$\widehat{\mathscr{M}}^{\downarrow}[\![\mathtt{\{\ Sl\ \}}]\!] \triangleq \widehat{\mathscr{M}}^{\downarrow}[\![\mathtt{Sl}]\!] \quad (37) \quad \square$$

To model check $\widehat{\mathscr{M}}[\![\mathtt{Sl}]\!]\langle \underline{\varrho},\, \mathtt{R}\rangle$ statement lists $\mathtt{Sl} ::= \mathtt{Sl}'\ \mathtt{S}$, we will need to model check the concatenation of traces of $\mathtt{Sl}'$ and $\mathtt{S}$ using the following lemma. This lemma will also be useful to handle the concatenation of traces in iterates.

Lemma 18. $\mathscr{M}^t\langle \underline{\varrho},\, \mathtt{R}\rangle(\pi \cdot \pi') = \langle \mathtt{tt},\, \mathtt{R}'\rangle \Leftrightarrow (\exists \mathtt{R}'' \in \mathbb{R}\ .\ \mathscr{M}^t\langle \underline{\varrho},\, \mathtt{R}\rangle(\pi) = \langle \mathtt{tt},\, \mathtt{R}''\rangle \wedge \mathscr{M}^t\langle \underline{\varrho},\, \mathtt{R}''\rangle(\pi') = \langle \mathtt{tt},\, \mathtt{R}'\rangle)$.

Proof (of Lem. 18). Since the case of empty traces is trivial by $\mathscr{M}^t\langle \underline{\varrho},\, \mathtt{R}\rangle\mathfrak{d} \triangleq \langle \mathtt{tt},\, \mathtt{R}\rangle$ in (18), we can assume that $\pi = \langle \ell_1,\, \rho_1\rangle \ldots \langle \ell_p,\, \rho_p\rangle$, $\pi' = \langle \ell_p,\, \rho_p\rangle \ldots \langle \ell_\ell,\, \rho_\ell\rangle$ with $1 \leqslant p \leqslant \ell$ so that, by def. of trace concatenation, $\pi \cdot \pi' = \langle \ell_1,\, \rho_1\rangle \ldots \langle \ell_p,\, \rho_p\rangle \ldots \langle \ell_\ell,\, \rho_\ell\rangle$.

Let us consider the decomposition of $\mathtt{R} \eqslantless \mathtt{L}_1 : \mathtt{B}_1 \bullet \mathtt{R}_1 \eqslantless \mathtt{L}_1 : \mathtt{B}_1 \bullet \mathtt{L}_2 : \mathtt{B}_2 \bullet \mathtt{R}_2 \eqslantless \ldots \eqslantless \mathtt{L}_1 : \mathtt{B}_1 \bullet \mathtt{L}_2 : \mathtt{B}_2 \bullet \ldots \bullet \mathtt{L}_n : \mathtt{B}_n \bullet \mathtt{R}_n$ of Lem. 14 where, by convention, $n \in \mathbb{N} \setminus \{0\}$ and $\mathtt{R}_n = \varepsilon$ or $n = +\infty$ and $\mathtt{R}_n$ is undefined.

Since $\mathscr{M}^t\langle \underline{\varrho}, \mathtt{R}\rangle(\pi \frown \pi') = \langle \mathtt{tt}, \mathtt{R}'\rangle$, we are in case (14.1) since the specification $\mathtt{R}$ is satisfied, in which case

$$\mathsf{let}\, m = \min(n, \ell)\ \mathsf{in}\ \forall i \in [1, m]\ .\ \langle \underline{\varrho}, \langle \ell_i, \rho_i\rangle\rangle \in \mathscr{S}^r[\![\mathtt{L}_i : \mathtt{B}_i]\!] \text{ and so } \mathtt{R}_m = \mathtt{R}'. \quad \text{(a)}$$

There are three cases.

- If $n \leqslant p \leqslant \ell$ in (a), then $m = n$ so $\mathtt{R}_n = \mathtt{R}' = \varepsilon$. First (b), we have $\forall i \in [1, p] = [1, \min(p, \ell)]\ .\ \langle \underline{\varrho}, \langle \ell_i, \rho_i\rangle\rangle \in \mathscr{S}^r[\![\mathtt{L}_i : \mathtt{B}_i]\!]$ and so $\mathscr{M}^t\langle \underline{\varrho}, \mathtt{R}\rangle\pi = \langle \mathtt{tt}, \mathtt{R}_p\rangle$ with $\mathtt{R}'' = \mathtt{R}_p$, which in this case is ε. Second (c), by (18), $\mathscr{M}^t\langle \underline{\varrho}, \mathtt{R}''\rangle(\pi') = \mathscr{M}^t\langle \underline{\varrho}, \varepsilon\rangle(\pi') = \langle \mathtt{tt}, \varepsilon\rangle = \langle \mathtt{tt}, \mathtt{R}'\rangle$, proving $\Rightarrow$. Conversely, $\mathscr{M}^t\langle \underline{\varrho}, \mathtt{R}\rangle(\pi) = \langle \mathtt{tt}, \mathtt{R}''\rangle$ implies (b) and $\mathscr{M}^t\langle \underline{\varrho}, \mathtt{R}''\rangle(\pi') = \langle \mathtt{tt}, \mathtt{R}'\rangle$ implies (c) which imply (a), hence $\mathscr{M}^t\langle \underline{\varrho}, \mathtt{R}\rangle(\pi \frown \pi') = \langle \mathtt{tt}, \mathtt{R}'\rangle$, proving $\Leftarrow$.
- If $p \leqslant n \leqslant \ell$ in (a), then again $m = n$ so $\mathtt{R}_n = \mathtt{R}' = \varepsilon$ hence (b) holds. Moreover, we have (d) $\forall i \in [p, n]\ .\ \langle \underline{\varrho}, \langle \ell_i, \rho_i\rangle\rangle \in \mathscr{S}^r[\![\mathtt{L}_i : \mathtt{B}_i]\!]$ and so $\mathscr{M}^t\langle \underline{\varrho}, \mathtt{R}_p\rangle(\pi') = \langle \mathtt{tt}, \varepsilon\rangle$, proving $\Leftarrow$ for $\mathtt{R}'' = \mathtt{R}_p$. Conversely, $\mathscr{M}^t\langle \underline{\varrho}, \mathtt{R}\rangle(\pi) = \langle \mathtt{tt}, \mathtt{R}''\rangle$ implies (b) and $\mathscr{M}^t\langle \underline{\varrho}, \mathtt{R}''\rangle(\pi') = \langle \mathtt{tt}, \mathtt{R}'\rangle$ implies (d) which imply (a), hence $\mathscr{M}^t\langle \underline{\varrho}, \mathtt{R}\rangle(\pi \frown \pi') = \langle \mathtt{tt}, \mathtt{R}'\rangle$, proving $\Leftarrow$.
- If $p \leqslant \ell \leqslant n$ in (a), then $m = \ell$ with (e) $\forall i \in [1, \ell]\ .\ \langle \underline{\varrho}, \langle \ell_i, \rho_i\rangle\rangle \in \mathscr{S}^r[\![\mathtt{L}_i : \mathtt{B}_i]\!]$ and so $\mathtt{R}_\ell = \mathtt{R}'$. This implies $\mathscr{M}^t\langle \underline{\varrho}, \mathtt{R}\rangle(\pi) = \langle \mathtt{tt}, \mathtt{R}_p\rangle$ since (f) $\forall i \in [1, p]\ .\ \langle \underline{\varrho}, \langle \ell_i, \rho_i\rangle\rangle \in \mathscr{S}^r[\![\mathtt{L}_i : \mathtt{B}_i]\!]$ so $\mathtt{R}'' = \mathtt{R}_p$ and $\mathscr{M}^t\langle \underline{\varrho}, \mathtt{R}_p\rangle(\pi') = \langle \mathtt{tt}, \mathtt{R}_\ell\rangle$ since (g) $\forall i \in [p, \ell]\ .\ \langle \underline{\varrho}, \langle \ell_i, \rho_i\rangle\rangle \in \mathscr{S}^r[\![\mathtt{L}_i : \mathtt{B}_i]\!]$ so $\mathtt{R}' = \mathtt{R}_\ell$, proving $\Rightarrow$. Conversely, (f) and (g) imply (a), hence $\Leftarrow$.

To abstract fixpoint definitions, we use the following classical transfer lemma.

Lemma 19 (exact iterates abstraction). Assume $\langle C, \sqsubseteq\rangle$ and $\langle \mathcal{A}, \preccurlyeq\rangle$ are posets, $\bot$ is the infimum of $\langle C, \sqsubseteq\rangle$, $f \in C \to C$, the lub $\bigsqcup_{n\in\mathbb{N}} f^n(\bot)$ exists in $\langle C, \sqsubseteq\rangle$ such that $\mathsf{lfp}^{\sqsubseteq} f = \bigsqcup_{n\in\mathbb{N}} f^n(\bot)$, $\langle C, \sqsubseteq\rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle \mathcal{A}, \preccurlyeq\rangle$ is a Galois connection, $\overline{f} \in \mathcal{A} \to \mathcal{A}$, and we have the commutation condition $\forall n \in \mathbb{N}\ .\ \alpha(f^{n+1}(\bot)) = \overline{f}(\alpha(f^n(\bot)))$. Then the lub $\bigvee_{n\in\mathbb{N}} \overline{f}^n(\alpha(\bot))$ exists in $\langle \mathcal{A}, \preccurlyeq\rangle$ such that $\alpha(\mathsf{lfp}^{\sqsubseteq} f) = \bigvee_{n\in\mathbb{N}} \overline{f}^n(\alpha(\bot))$.

Note that if $\overline{f}$ is increasing and $\langle \mathcal{A}, \preccurlyeq\rangle$ is a cpo then $\alpha(\mathsf{lfp}^{\sqsubseteq} f) = \mathsf{lfp}^{\preccurlyeq} \overline{f}$.

We can now provide an a posteriori proof of Th. 16 (although the proof was done a priori to discover Def. 17 by calculational design).

Proof (of Th. 16). The proof is by calculational design and proceeds by structural induction on the program component $\mathtt{S}$.

— In case (27) of an empty temporal specification ε, we have

$\mathscr{M}^{\natural}[\![\mathtt{S}]\!]\langle \underline{\varrho},\ \varepsilon\rangle$

$\triangleq\ \mathscr{M}^{\natural}\langle \underline{\varrho},\ \varepsilon\rangle(\mathscr{S}^{*}[\![\mathtt{S}]\!])$ ⟅(20)⟆

$= \{\langle \pi,\ \mathtt{R}'\rangle \mid \pi \in \mathscr{S}^{*}[\![\mathtt{S}]\!] \wedge \langle \mathtt{tt},\ \mathtt{R}'\rangle = \mathscr{M}^{t}\langle \rho,\ \varepsilon\rangle\pi\}$ ⟅(19)⟆

$= \{\langle \pi,\ \varepsilon\rangle \mid \pi \in \mathscr{S}^{*}[\![\mathtt{S}]\!]\}$ ⟅since $\mathscr{M}^{t}\langle \underline{\varrho},\ \varepsilon\rangle\pi \triangleq \langle \mathtt{tt},\ \varepsilon\rangle$ by (18)⟆

$\triangleq\ \widehat{\mathscr{M}}^{\natural}[\![\mathtt{S}]\!]\langle \underline{\varrho},\ \varepsilon\rangle$ ⟅(27)⟆

— In case (28) of a statement list $\mathtt{Sl} ::= \mathtt{Sl}'\ \mathtt{S}$

$\mathscr{M}^{\natural}[\![\mathtt{Sl}]\!]\ \langle \underline{\varrho},\ \mathtt{R}\rangle$

$= \mathscr{M}^{\natural}\langle \underline{\varrho},\ \mathtt{R}\rangle(\mathscr{S}^{*}[\![\mathtt{Sl}]\!])$ ⟅(20) ⟆

$= \{\langle \pi,\ \mathtt{R}'\rangle \mid \pi \in \mathscr{S}^{*}[\![\mathtt{S}]\!] \wedge \langle \mathtt{tt},\ \mathtt{R}'\rangle = \mathscr{M}^{t}\langle \underline{\varrho},\ \mathtt{R}\rangle\pi\}$ ⟅(19)⟆

$= \{\langle \pi,\ \mathtt{R}'\rangle \mid \pi \in \mathscr{S}^{*}[\![\mathtt{Sl}'\,]\!] \cup \{\pi \cdot \langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle \cdot \pi' \mid \pi \cdot \langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle \in \mathscr{S}^{*}[\![\mathtt{Sl}'\,]\!] \wedge \langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle \cdot \pi' \in \mathscr{S}^{*}[\![\mathtt{S}]\!]\} \wedge \langle \mathtt{tt},\ \mathtt{R}'\rangle = \mathscr{M}^{t}\langle \underline{\varrho},\ \mathtt{R}\rangle\pi\}$ ⟅(7)⟆

$= \{\langle \pi,\ \mathtt{R}'\rangle \mid \pi \in \mathscr{S}^{*}[\![\mathtt{Sl}'\,]\!] \wedge \langle \mathtt{tt},\ \mathtt{R}'\rangle = \mathscr{M}^{t}\langle \underline{\varrho},\ \mathtt{R}\rangle\pi\} \cup$
$\{\langle \pi,\ \mathtt{R}'\rangle \mid \pi \in \{\pi' \cdot \langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle \cdot \pi'' \mid \pi' \cdot \langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle \in \mathscr{S}^{*}[\![\mathtt{Sl}'\,]\!] \wedge \langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle \cdot \pi'' \in \mathscr{S}^{*}[\![\mathtt{S}]\!]\} \wedge \langle \mathtt{tt},\ \mathtt{R}'\rangle = \mathscr{M}^{t}\langle \underline{\varrho},\ \mathtt{R}\rangle\pi\}$ ⟅def. ∪ and ∈⟆

$= \mathscr{M}^{\natural}[\![\mathtt{Sl}'\,]\!]\ \langle \underline{\varrho},\ \mathtt{R}\rangle \cup$
$\{\langle \pi' \cdot \langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle \cdot \pi'',\ \mathtt{R}'\rangle \mid \pi' \cdot \langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle \in \mathscr{S}^{*}[\![\mathtt{Sl}'\,]\!] \wedge \langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle \cdot \pi'' \in \mathscr{S}^{*}[\![\mathtt{S}]\!] \wedge \langle \mathtt{tt},\ \mathtt{R}'\rangle = \mathscr{M}^{t}\langle \underline{\varrho},\ \mathtt{R}\rangle(\pi' \cdot \langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle \cdot \pi'')\}$ ⟅(19) and (7), def. ∈⟆

$= \mathscr{M}^{\natural}[\![\mathtt{Sl}'\,]\!]\ \langle \underline{\varrho},\ \mathtt{R}\rangle \cup$
$\{\langle \pi' \cdot \langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle \cdot \pi'',\ \mathtt{R}'\rangle \mid \pi' \cdot \langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle \in \mathscr{S}^{*}[\![\mathtt{Sl}'\,]\!] \wedge \langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle \cdot \pi'' \in \mathscr{S}^{*}[\![\mathtt{S}]\!] \wedge (\exists \mathtt{R}'' \in \mathbb{R}\ .\ \mathscr{M}^{t}\langle \underline{\varrho},\ \mathtt{R}\rangle(\pi' \cdot \langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle) = \langle \mathtt{tt},\ \mathtt{R}''\rangle \wedge \mathscr{M}^{t}\langle \underline{\varrho},\ \mathtt{R}''\rangle(\langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle \cdot \pi'') = \langle \mathtt{tt},\ \mathtt{R}'\rangle)\}$
⟅Lem. 18⟆

$= \mathscr{M}^{\natural}[\![\mathtt{Sl}'\,]\!]\ \langle \underline{\varrho},\ \mathtt{R}\rangle \cup$
$\{\langle \pi' \cdot \langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle \cdot \pi'',\ \mathtt{R}'\rangle \mid \langle \pi' \cdot \langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle,\ \mathtt{R}''\rangle \in \mathscr{M}^{\natural}\langle \underline{\varrho},\ \mathtt{R}\rangle(\mathscr{S}^{*}[\![\mathtt{Sl}'\,]\!]) \wedge \langle \langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle \cdot \pi'',\ \mathtt{R}'\rangle \in \mathscr{M}^{\natural}\langle \underline{\varrho},\ \mathtt{R}''\rangle(\mathscr{S}^{*}[\![\mathtt{S}]\!])\}$ ⟅(19)⟆

$= \mathscr{M}^{\natural}[\![\mathtt{Sl}'\,]\!]\ \langle \underline{\varrho},\ \mathtt{R}\rangle \cup \{\langle \pi \cdot \langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle \cdot \pi',\ \mathtt{R}'\rangle \mid \langle \pi \cdot \langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle,\ \mathtt{R}''\rangle \in \mathscr{M}^{t}[\![\mathtt{Sl}'\,]\!]\langle \underline{\varrho},\ \mathtt{R}\rangle \wedge \langle \langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle \cdot \pi',\ \mathtt{R}'\rangle \in \mathscr{M}^{t}[\![\mathtt{S}]\!]\langle \underline{\varrho},\ \mathtt{R}''\rangle\}$ ⟅(20)⟆

$\mathscr{M}^{\natural}[\![\mathtt{Sl}'\,]\!]\ \langle \underline{\varrho},\ \mathtt{R}\rangle \cup \{\langle \pi \cdot \langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle \cdot \pi',\ \mathtt{R}'\rangle \mid \langle \pi \cdot \langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle,\ \mathtt{R}''\rangle \in \widehat{\mathscr{M}}^{\natural}[\![\mathtt{Sl}'\,]\!]\langle \underline{\varrho},\ \mathtt{R}\rangle \wedge \langle \langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle \cdot \pi',\ \mathtt{R}'\rangle \in \widehat{\mathscr{M}}^{\natural}[\![\mathtt{S}]\!]\langle \underline{\varrho},\ \mathtt{R}''\rangle\}$ ⟅ind. hyp.⟆

$\triangleq\ \widehat{\mathscr{M}}^{\natural}[\![\mathtt{Sl}]\!]\langle \underline{\varrho},\ \mathtt{R}\rangle$ ⟅(28) ⟆

— In case (29) of an empty statement list $\mathtt{Sl} ::= \epsilon$

$\mathscr{M}^{\natural}[\![\mathtt{Sl}]\!]\ \langle \underline{\varrho},\ \mathtt{R}\rangle$

$= \mathscr{M}^{\natural}\langle \underline{\varrho},\ \mathtt{R}\rangle(\mathscr{S}^{*}[\![\mathtt{Sl}]\!])$ ⟅(20) ⟆

$= \{\langle \pi,\ \mathtt{R}' \rangle \mid \pi \in \mathcal{S}^*[\![\mathtt{Sl}]\!] \wedge \langle \mathtt{tt},\ \mathtt{R}' \rangle = \mathcal{M}^t \langle \underline{\varrho},\ \mathtt{R} \rangle \pi\}$ ⟅(19)⟆

$= \{\langle \pi,\ \mathtt{R}' \rangle \mid \pi \in \{\langle \mathsf{at}[\![\mathtt{Sl}]\!],\ \rho \rangle \mid \rho \in \mathbb{E}\mathfrak{v}\} \wedge \langle \mathtt{tt},\ \mathtt{R}' \rangle = \mathcal{M}^t \langle \underline{\varrho},\ \mathtt{R} \rangle \pi\}$ ⟅(6)⟆

$= \{\langle \langle \mathsf{at}[\![\mathtt{Sl}]\!],\ \rho \rangle,\ \mathtt{R}' \rangle \mid \rho \in \mathbb{E}\mathfrak{v} \wedge \langle \mathtt{tt},\ \mathtt{R}' \rangle = \mathcal{M}^t \langle \underline{\varrho},\ \mathtt{R} \rangle (\langle \mathsf{at}[\![\mathtt{Sl}]\!],\ \rho \rangle)\}$ ⟅def. ∈⟆

$= \{\langle \langle \mathsf{at}[\![\mathtt{Sl}]\!],\ \rho \rangle,\ \mathtt{R}' \rangle \mid \rho \in \mathbb{E}\mathfrak{v} \wedge \langle \mathtt{L} : \mathtt{B},\ \mathtt{R}' \rangle = \mathsf{fstnxt}(\mathtt{R}) \wedge \langle \underline{\varrho},\ \langle \mathsf{at}[\![\mathtt{Sl}]\!],\ \rho \rangle \rangle \in \mathcal{S}^r[\![\mathtt{L} : \mathtt{B}]\!]\}$
⟅(18) with $\mathcal{M}^t \langle \underline{\varrho},\ \mathtt{R}' \rangle \ni = \langle \mathtt{tt},\ \mathtt{R}' \rangle$⟆

$= \widehat{\mathcal{M}}^{\natural}[\![\mathtt{Sl}]\!] \langle \underline{\varrho},\ \mathtt{R} \rangle$ ⟅(29)⟆

— In case (35) of a skip statement $\mathtt{S} ::= \mathtt{;}$

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

$= \{\langle \pi,\ \mathtt{R}' \rangle \mid \pi \in \mathcal{S}^*[\![\mathtt{S}]\!] \wedge \langle \mathtt{tt},\ \mathtt{R}' \rangle = \mathcal{M}^t \langle \underline{\varrho},\ \mathtt{R} \rangle \pi\}$ ⟅(20) and (19)⟆

$= \{\langle \pi,\ \mathtt{R}' \rangle \mid \pi \in \{\langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho \rangle \mid \rho \in \mathbb{E}\mathfrak{v}\} \wedge \langle \mathtt{tt},\ \mathtt{R}' \rangle = \mathcal{M}^t \langle \underline{\varrho},\ \mathtt{R} \rangle \pi\}$ ⟅(10)⟆

$= \{\langle \langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho \rangle,\ \mathtt{R}' \rangle \mid \rho \in \mathbb{E}\mathfrak{v} \wedge \langle \mathtt{tt},\ \mathtt{R}' \rangle = \mathcal{M}^t \langle \underline{\varrho},\ \mathtt{R} \rangle (\langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho \rangle)\}$ ⟅def. ∈⟆

$= \{\langle \langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho \rangle,\ \mathtt{R}' \rangle \mid \rho \in \mathbb{E}\mathfrak{v} \wedge \langle \mathtt{L} : \mathtt{B},\ \mathtt{R}' \rangle = \mathsf{fstnxt}(\mathtt{R}) \wedge \langle \underline{\varrho},\ \langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho \rangle \rangle \in \mathcal{S}^r[\![\mathtt{L} : \mathtt{B}]\!]\}$
⟅(18) with $\mathcal{M}^t \langle \underline{\varrho},\ \mathtt{R}' \rangle \ni = \langle \mathtt{tt},\ \mathtt{R}' \rangle$⟆

$= \widehat{\mathcal{M}}^{\natural}[\![\mathtt{S}]\!] \langle \underline{\varrho},\ \mathtt{R} \rangle$ ⟅(35)⟆

— In case (31) of a conditional statement $\mathtt{S} ::= \mathtt{if}\ ^{\ell}\ \mathtt{(B)}\ \mathtt{S}_t$

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

$= \mathcal{M}^{\natural} \langle \underline{\varrho},\ \mathtt{R} \rangle (\mathcal{S}^*[\![\mathtt{S}]\!])$ ⟅(20) ⟆

$= \{\langle \pi,\ \mathtt{R}' \rangle \mid \pi \in \mathcal{S}^*[\![\mathtt{S}]\!] \wedge \langle \mathtt{tt},\ \mathtt{R}' \rangle = \mathcal{M}^t \langle \underline{\varrho},\ \mathtt{R} \rangle \pi\}$ ⟅(19)⟆

$= \{\langle \pi,\ \mathtt{R}' \rangle \mid \pi \in \{\langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho \rangle \mid \rho \in \mathbb{E}\mathfrak{v}\} \wedge \langle \mathtt{tt},\ \mathtt{R}' \rangle = \mathcal{M}^t \langle \underline{\varrho},\ \mathtt{R} \rangle \pi\} \cup$
$\{\langle \pi,\ \mathtt{R}' \rangle \mid \pi \in \{\langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho \rangle \langle \mathsf{aft}[\![\mathtt{S}]\!],\ \rho \rangle \mid \mathcal{B}[\![\mathtt{B}]\!]\ \rho = \mathtt{ff}\} \wedge \langle \mathtt{tt},\ \mathtt{R}' \rangle = \mathcal{M}^t \langle \underline{\varrho},\ \mathtt{R} \rangle \pi\} \cup$
$\{\langle \pi,\ \mathtt{R}' \rangle \mid \pi \in \{\langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho \rangle \langle \mathsf{at}[\![\mathtt{S}_t]\!],\ \rho \rangle \fatsemi \pi_2 \mid \mathcal{B}[\![\mathtt{B}]\!]\ \rho = \mathtt{tt} \wedge \pi_2 \in \mathcal{S}^*[\![\mathtt{S}_t]\!]\} \wedge \langle \mathtt{tt},$
$\mathtt{R}' \rangle = \mathcal{M}^t \langle \underline{\varrho},\ \mathtt{R} \rangle \pi\}$ ⟅(4) and def. ∪⟆

We go on separately for each of the above three terms.

— The first term was already handled above in the proof of (35) for the a skip statement $\mathtt{S} ::= \mathtt{;}$.

$\{\langle \pi,\ \mathtt{R}' \rangle \mid \pi \in \{\langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho \rangle \mid \rho \in \mathbb{E}\mathfrak{v}\} \wedge \langle \mathtt{tt},\ \mathtt{R}' \rangle = \mathcal{M}^t \langle \underline{\varrho},\ \mathtt{R} \rangle \pi\}$

$= \mathsf{let}\ \langle \mathtt{L} : \mathtt{B},\ \mathtt{R}' \rangle = \mathsf{fstnxt}(\mathtt{R})\ \mathsf{in}\ \{\langle \langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho \rangle,\ \mathtt{R}' \rangle \mid \langle \underline{\varrho},\ \langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho \rangle \rangle \in \mathcal{S}^r[\![\mathtt{L} : \mathtt{B}]\!]\}$

— The second term is

$\{\langle \pi,\ \mathtt{R}' \rangle \mid \pi \in \{\langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho \rangle \langle \mathsf{aft}[\![\mathtt{S}]\!],\ \rho \rangle \mid \mathcal{B}[\![\mathtt{B}]\!]\ \rho = \mathtt{ff}\} \wedge \langle \mathtt{tt},\ \mathtt{R}' \rangle = \mathcal{M}^t \langle \underline{\varrho},\ \mathtt{R} \rangle \pi\}$

$= \{\langle \langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho \rangle \langle \mathsf{aft}[\![\mathtt{S}]\!],\ \rho \rangle,\ \mathtt{R}' \rangle \mid \mathcal{B}[\![\mathtt{B}]\!]\ \rho = \mathtt{ff} \wedge \langle \mathtt{tt},\ \mathtt{R}' \rangle = \mathcal{M}^t \langle \underline{\varrho},\ \mathtt{R} \rangle (\langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho \rangle \langle \mathsf{aft}[\![\mathtt{S}]\!],$
$\rho \rangle)\}$ ⟅def. ∈⟆

By (18), there are two subcases. If $\mathtt{R} = \varepsilon$, this is

$= \{\langle \langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho \rangle \langle \mathsf{aft}[\![\mathtt{S}]\!],\ \rho \rangle,\ \varepsilon \rangle \mid \mathtt{R} \in \mathbb{R}_{\varepsilon} \wedge \mathcal{B}[\![\mathtt{B}]\!]\ \rho = \mathtt{ff}\}$ ⟅(18)⟆

This term appears as (27). Otherwise $\mathtt{R} \neq \varepsilon$, and this is

$= \{\langle\langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle\langle \mathsf{aft}[\![\mathsf{S}]\!],\ \rho\rangle,\ \mathsf{R}'\rangle \mid \mathsf{R} \notin \mathcal{R}_\varepsilon \wedge \mathscr{B}[\![\mathsf{B}]\!]\ \rho = \mathsf{ff} \wedge \langle \mathsf{tt},\ \mathsf{R}'\rangle = \mathsf{let}\,\langle \mathsf{L} : \mathsf{B},\ \mathsf{R}'\rangle = \mathsf{fstnxt}(\mathsf{R})\ \mathsf{in}\ (\!|\ \langle \underline{\varrho},\ \langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle\rangle \in \mathcal{S}^r[\![\mathsf{L} : \mathsf{B}]\!] \ ?\ \mathscr{M}^t\langle \underline{\varrho},\ \mathsf{R}'\rangle\langle \mathsf{aft}[\![\mathsf{S}]\!],\ \rho\rangle \ :\ \langle \mathsf{ff},\ \mathsf{R}'\rangle\ |\!)\}$ ⦇(18)⦈

$= \{\langle\langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle\langle \mathsf{aft}[\![\mathsf{S}]\!],\ \rho\rangle,\ \mathsf{R}'\rangle \mid \mathsf{R} \notin \mathcal{R}_\varepsilon \wedge \mathscr{B}[\![\mathsf{B}]\!]\rho = \mathsf{ff} \wedge \mathsf{let}\,\langle \mathsf{L} : \mathsf{B},\ \mathsf{R}'\rangle = \mathsf{fstnxt}(\mathsf{R})\ \mathsf{in}\ \langle \underline{\varrho},\ \langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle\rangle \in \mathcal{S}^r[\![\mathsf{L} : \mathsf{B}]\!] \wedge \langle \mathsf{tt},\ \mathsf{R}'\rangle = \mathscr{M}^t\langle \underline{\varrho},\ \mathsf{R}'\rangle\langle \mathsf{aft}[\![\mathsf{S}]\!],\ \rho\rangle\}$ ⦇def. conditional⦈

By (18), there are two subsubcases. If $\mathsf{R}' = \varepsilon$, this is

$= \{\langle\langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle\langle \mathsf{aft}[\![\mathsf{S}]\!],\ \rho\rangle,\ \varepsilon\rangle \mid \mathsf{R} \notin \mathcal{R}_\varepsilon \wedge \mathscr{B}[\![\mathsf{B}]\!]\ \rho = \mathsf{ff} \wedge \langle \mathsf{L} : \mathsf{B},\ \varepsilon\rangle = \mathsf{fstnxt}(\mathsf{R}) \wedge \langle \underline{\varrho},\ \langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle\rangle \wedge \mathcal{S}^r[\![\mathsf{L} : \mathsf{B}]\!]\}$ ⦇(18)⦈

$= \{\langle\langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle\langle \mathsf{aft}[\![\mathsf{S}]\!],\ \rho\rangle,\ \varepsilon\rangle \mid \mathsf{R} \notin \mathcal{R}_\varepsilon \wedge \mathscr{B}[\![\mathsf{B}]\!]\rho = \mathsf{ff} \wedge \langle \mathsf{L}' : \mathsf{B}',\ \mathsf{R}'\rangle = \mathsf{fstnxt}(\mathsf{R}) \wedge \mathsf{R}' \in \mathcal{R}_\varepsilon \wedge \langle \underline{\varrho},\ \langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle\rangle \in \mathcal{S}^r[\![\mathsf{L}' : \mathsf{B}']\!]\}$

Otherwise $\mathsf{R} \neq \varepsilon$, and this is

$= \{\langle\langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle\langle \mathsf{aft}[\![\mathsf{S}]\!],\ \rho\rangle,\ \mathsf{R}''\rangle \mid \mathsf{R} \notin \mathcal{R}_\varepsilon \wedge \mathscr{B}[\![\mathsf{B}]\!]\ \rho = \mathsf{ff} \wedge \langle \mathsf{L} : \mathsf{B},\ \mathsf{R}'\rangle = \mathsf{fstnxt}(\mathsf{R}) \wedge \langle \underline{\varrho},\ \langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle\rangle \in \mathcal{S}^r[\![\mathsf{L} : \mathsf{B}]\!] \wedge \mathsf{R}' \notin \mathcal{R}_\varepsilon \wedge \langle \mathsf{L}'' : \mathsf{B}'',\ \mathsf{R}''\rangle = \mathsf{fstnxt}(\mathsf{R}') \wedge \langle \underline{\varrho},\ \langle \mathsf{aft}[\![\mathsf{S}]\!],\ \rho\rangle\rangle \in \mathcal{S}^r[\![\mathsf{L}'' : \mathsf{B}'']\!]\}$ ⦇(18)⦈

— The third term is

$\{\langle \pi,\ \mathsf{R}'\rangle \mid \pi \in \{\langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle\langle \mathsf{at}[\![\mathsf{S}_t]\!],\ \rho\rangle \cdot \pi_2 \mid \mathscr{B}[\![\mathsf{B}]\!]\ \rho = \mathsf{tt} \wedge \pi_2 \in \mathcal{S}^*[\![\mathsf{S}_t]\!]\} \wedge \langle \mathsf{tt},\ \mathsf{R}'\rangle = \mathscr{M}^t\langle \underline{\varrho},\ \mathsf{R}\rangle\pi\}$

$= \{\langle\langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle\langle \mathsf{at}[\![\mathsf{S}_t]\!],\ \rho\rangle\pi_2,\ \mathsf{R}'\rangle \mid \mathscr{B}[\![\mathsf{B}]\!]\ \rho = \mathsf{tt} \wedge \langle \mathsf{at}[\![\mathsf{S}_t]\!],\ \rho\rangle\pi_2 \in \mathcal{S}^*[\![\mathsf{S}_t]\!] \wedge \langle \mathsf{tt},\ \mathsf{R}'\rangle = \mathscr{M}^t\langle \underline{\varrho},\ \mathsf{R}\rangle(\langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle\langle \mathsf{at}[\![\mathsf{S}_t]\!],\ \rho\rangle\pi_2)\}$ ⦇def. ∈ and ·⦈

By (18), there are two subases. If $\mathsf{R} = \varepsilon$, this is

$= \{\langle\langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle\langle \mathsf{at}[\![\mathsf{S}_t]\!],\ \rho\rangle\pi_2,\ \varepsilon\rangle \mid \mathsf{R} \in \mathcal{R}_\varepsilon \wedge \mathscr{B}[\![\mathsf{B}]\!]\ \rho = \mathsf{tt} \wedge \langle \mathsf{at}[\![\mathsf{S}_t]\!],\ \rho\rangle\pi_2 \in \mathcal{S}^*[\![\mathsf{S}_t]\!]\}$ ⦇(18)⦈

This term is incorporated in (27). Otherwise $\mathsf{R} \neq \varepsilon$, and this is

$= \{\langle\langle \mathsf{at}[\![\mathsf{S}]\!], \rho\rangle\langle \mathsf{at}[\![\mathsf{S}_t]\!], \rho\rangle\pi_2, \mathsf{R}''\rangle \mid \mathsf{R} \notin \mathcal{R}_\varepsilon \wedge \mathscr{B}[\![\mathsf{B}]\!]\rho = \mathsf{tt} \wedge \langle \mathsf{at}[\![\mathsf{S}_t]\!], \rho\rangle\pi_2 \in \mathcal{S}^*[\![\mathsf{S}_t]\!] \wedge \langle \mathsf{L} : \mathsf{B},\ \mathsf{R}'\rangle = \mathsf{fstnxt}(\mathsf{R}) \wedge \langle \underline{\varrho},\ \langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle\rangle \in \mathcal{S}^r[\![\mathsf{L} : \mathsf{B}]\!] \wedge \langle \mathsf{tt},\ \mathsf{R}''\rangle = \mathscr{M}^t\langle \underline{\varrho},\ \mathsf{R}'\rangle(\langle \mathsf{at}[\![\mathsf{S}_t]\!],\ \rho\rangle\pi_2)\}$ ⦇(18)⦈

$= \{\langle\langle \mathsf{at}[\![\mathsf{S}]\!], \rho\rangle\langle \mathsf{at}[\![\mathsf{S}_t]\!], \rho\rangle\pi_2, \mathsf{R}''\rangle \mid \mathsf{R} \notin \mathcal{R}_\varepsilon \wedge \mathscr{B}[\![\mathsf{B}]\!]\rho = \mathsf{tt} \wedge \langle \mathsf{L} : \mathsf{B},\ \mathsf{R}'\rangle = \mathsf{fstnxt}(\mathsf{R}) \wedge \langle \underline{\varrho},\ \langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle\rangle \in \mathcal{S}^r[\![\mathsf{L} : \mathsf{B}]\!] \wedge \langle\langle \mathsf{at}[\![\mathsf{S}_t]\!],\ \rho\rangle\pi_2,\ \mathsf{R}''\rangle \in \{\langle \pi,\ \mathsf{R}''\rangle \mid \pi \in \mathcal{S}^*[\![\mathsf{S}_t]\!] \wedge \langle \mathsf{tt},\ \mathsf{R}''\rangle = \mathscr{M}^t\langle \underline{\varrho},\ \mathsf{R}'\rangle\pi\}\}$ ⦇def. ∈⦈

$= \{\langle\langle \mathsf{at}[\![\mathsf{S}]\!], \rho\rangle\langle \mathsf{at}[\![\mathsf{S}_t]\!], \rho\rangle\pi_2, \mathsf{R}''\rangle \mid \mathsf{R} \notin \mathcal{R}_\varepsilon \wedge \mathscr{B}[\![\mathsf{B}]\!]\rho = \mathsf{tt} \wedge \langle \mathsf{L} : \mathsf{B},\ \mathsf{R}'\rangle = \mathsf{fstnxt}(\mathsf{R}) \wedge \langle \underline{\varrho},\ \langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle\rangle \in \mathcal{S}^r[\![\mathsf{L} : \mathsf{B}]\!] \wedge \langle\langle \mathsf{at}[\![\mathsf{S}_t]\!],\ \rho\rangle\pi_2,\ \mathsf{R}''\rangle \in \mathscr{M}^\natural[\![\mathsf{S}_t]\!]\langle \underline{\varrho},\ \mathsf{R}'\rangle\}$ ⦇(20) and (19)⦈

— Grouping all cases and subcases together, we get (31) and (27) when R is empty. The case (36) is similar.

— In case (33) of an iteration statement $\mathsf{S} ::= \mathtt{while}^{\ell}\ \mathtt{(B)}\ \mathsf{S}_b$, we apply Lem. 19 so we have to calculate the abstract transformer that satisfies the commutation property for an iterate X of the concrete transformer $\mathscr{F}^*[\![\mathsf{S}]\!]$ (which traces must be of the form $\pi\langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle$).

$\mathscr{M}^\natural\langle \underline{\varrho},\, \mathsf{R}\rangle(\mathscr{F}^*[\![\mathsf{S}]\!]\, X)$

$= \mathscr{M}^\natural\langle \underline{\varrho},\, \mathsf{R}\rangle(\{\langle \ell,\, \rho\rangle \mid \rho \in \mathbb{E}\mathsf{v}\} \cup \{\pi_2\langle \ell',\, \rho\rangle\langle \mathsf{aft}[\![\mathsf{S}]\!],\, \rho\rangle \mid \pi_2\langle \ell',\, \rho\rangle \in X \wedge \mathscr{B}[\![\mathsf{B}]\!]\, \rho = \mathsf{ff} \wedge \ell' = \ell\} \cup \{\pi_2\langle \ell',\, \rho\rangle\langle \mathsf{at}[\![\mathsf{S}_b]\!],\, \rho\rangle \cdot \pi_3 \mid \pi_2\langle \ell',\, \rho\rangle \in X \wedge \mathscr{B}[\![\mathsf{B}]\!]\, \rho = \mathsf{tt} \wedge \langle \mathsf{at}[\![\mathsf{S}_b]\!],\, \rho\rangle \cdot \pi_3 \in \mathscr{S}^*[\![\mathsf{S}_b]\!] \wedge \ell' = \ell\})$ ⟅(8)⟆

$= \mathscr{M}^\natural\langle \underline{\varrho},\, \mathsf{R}\rangle(\{\langle \ell,\, \rho\rangle \mid \rho \in \mathbb{E}\mathsf{v}\}) \cup \mathscr{M}^\natural\langle \underline{\varrho},\, \mathsf{R}\rangle(\{\pi_2\langle \ell',\, \rho\rangle\langle \mathsf{aft}[\![\mathsf{S}]\!],\, \rho\rangle \mid \pi_2\langle \ell',\, \rho\rangle \in X \wedge \mathscr{B}[\![\mathsf{B}]\!]\, \rho = \mathsf{ff} \wedge \ell' = \ell\}) \cup \mathscr{M}^\natural\langle \underline{\varrho},\, \mathsf{R}\rangle(\{\pi_2\langle \ell',\, \rho\rangle\langle \mathsf{at}[\![\mathsf{S}_b]\!],\, \rho\rangle \cdot \pi_3 \mid \pi_2\langle \ell',\, \rho\rangle \in X \wedge \mathscr{B}[\![\mathsf{B}]\!]\, \rho = \mathsf{tt} \wedge \langle \mathsf{at}[\![\mathsf{S}_b]\!],\, \rho\rangle \cdot \pi_3 \in \mathscr{S}^*[\![\mathsf{S}_b]\!] \wedge \ell' = \ell\})$

⟅Galois connection (23), so that $\mathscr{M}^\natural\langle \underline{\varrho},\, \mathsf{R}\rangle$ preserves joins⟆

To avoid repeating (27), we assume that $\mathsf{R} \notin \mathscr{R}_\varepsilon$ so we can let $\langle \mathsf{L}' : \mathsf{B}',\, \mathsf{R}'\rangle = \mathsf{fstnxt}(\mathsf{R})$. There are three subcases.

— The first case is that of an observation of the execution that stops at loop entry $\ell = \mathsf{at}[\![\mathsf{S}]\!]$. This is similar to the above proof *e.g.* of (35) for a skip statement, and we get

$\mathscr{M}^\natural\langle \underline{\varrho},\, \mathsf{R}\rangle(\{\langle \mathsf{at}[\![\mathsf{S}]\!],\, \rho\rangle \mid \rho \in \mathbb{E}\mathsf{v}\}$

$= \{\langle\langle \mathsf{at}[\![\mathsf{S}]\!], \rho\rangle, \mathsf{R}'\rangle \mid \rho \in \mathbb{E}\mathsf{v} \wedge \langle \mathsf{L}' : \mathsf{B}', \mathsf{R}'\rangle = \mathsf{fstnxt}(\mathsf{R}) \wedge \langle \underline{\varrho}, \langle \mathsf{at}[\![\mathsf{S}]\!], \rho\rangle\rangle \in \mathscr{S}^r[\![\mathsf{L}' : \mathsf{B}']\!]\}$

— The second case is that of the loop exit

$\mathscr{M}^\natural\langle \underline{\varrho},\, \mathsf{R}\rangle(\{\pi_2\langle \mathsf{at}[\![\mathsf{S}]\!],\, \rho\rangle\langle \mathsf{aft}[\![\mathsf{S}]\!],\, \rho\rangle \mid \pi_2\langle \mathsf{at}[\![\mathsf{S}]\!],\, \rho\rangle \in X \wedge \mathscr{B}[\![\mathsf{B}]\!]\, \rho = \mathsf{ff}\})$

$= \{\langle \pi,\, \mathsf{R}'\rangle \mid \pi \in \{\pi_2\langle \mathsf{at}[\![\mathsf{S}]\!],\, \rho\rangle\langle \mathsf{aft}[\![\mathsf{S}]\!],\, \rho\rangle \mid \pi_2\langle \mathsf{at}[\![\mathsf{S}]\!],\, \rho\rangle \in X \wedge \mathscr{B}[\![\mathsf{B}]\!]\, \rho = \mathsf{ff}\} \wedge \langle \mathsf{tt},\, \mathsf{R}'\rangle = \mathscr{M}^t\langle \underline{\varrho},\, \mathsf{R}\rangle\pi\}$ ⟅(19)⟆

$= \{\langle \pi_2\langle \mathsf{at}[\![\mathsf{S}]\!],\, \rho\rangle\langle \mathsf{aft}[\![\mathsf{S}]\!],\, \rho\rangle,\, \mathsf{R}'\rangle \mid \pi_2\langle \mathsf{at}[\![\mathsf{S}]\!],\, \rho\rangle \in X \wedge \mathscr{B}[\![\mathsf{B}]\!]\, \rho = \mathsf{ff} \wedge \langle \mathsf{tt},\, \mathsf{R}'\rangle = \mathscr{M}^t\langle \underline{\varrho},\, \mathsf{R}\rangle(\pi_2\langle \mathsf{at}[\![\mathsf{S}]\!],\, \rho\rangle\langle \mathsf{aft}[\![\mathsf{S}]\!],\, \rho\rangle)\}$ ⟅def. ∈⟆

$= \{\langle \pi_2\langle \mathsf{at}[\![\mathsf{S}]\!],\, \rho\rangle\langle \mathsf{aft}[\![\mathsf{S}]\!],\, \rho\rangle,\, \mathsf{R}'\rangle \mid \pi_2\langle \mathsf{at}[\![\mathsf{S}]\!],\, \rho\rangle \in X \wedge \mathscr{B}[\![\mathsf{B}]\!]\, \rho = \mathsf{ff} \wedge \exists \mathsf{R}'' \in \mathscr{R}\,.\ \mathscr{M}^t\langle \underline{\varrho},\, \mathsf{R}\rangle(\pi_2\langle \mathsf{at}[\![\mathsf{S}]\!],\, \rho\rangle) = \langle \mathsf{tt},\, \mathsf{R}''\rangle \wedge \mathscr{M}^t\langle \underline{\varrho},\, \mathsf{R}''\rangle(\langle \mathsf{at}[\![\mathsf{S}]\!],\, \rho\rangle\langle \mathsf{aft}[\![\mathsf{S}]\!],\, \rho\rangle) = \langle \mathsf{tt},\, \mathsf{R}'\rangle\}$
⟅Lem. 18⟆

$= \{\langle \pi_2\langle \mathsf{at}[\![\mathsf{S}]\!],\, \rho\rangle\langle \mathsf{aft}[\![\mathsf{S}]\!],\, \rho\rangle,\, \mathsf{R}'\rangle \mid \langle \pi_2\langle \mathsf{at}[\![\mathsf{S}]\!],\, \rho\rangle,\, \mathsf{R}''\rangle \in \{\langle \pi,\, \mathsf{R}''\rangle \mid \pi \in X \wedge \langle \mathsf{tt},\, \mathsf{R}''\rangle = \mathscr{M}^t\langle \underline{\varrho},\, \mathsf{R}\rangle\pi\} \wedge \mathscr{B}[\![\mathsf{B}]\!]\, \rho = \mathsf{ff} \wedge \mathscr{M}^t\langle \underline{\varrho},\, \mathsf{R}''\rangle(\langle \mathsf{at}[\![\mathsf{S}]\!],\, \rho\rangle\langle \mathsf{aft}[\![\mathsf{S}]\!],\, \rho\rangle) = \langle \mathsf{tt},\, \mathsf{R}'\rangle\}$
⟅X is an iterate of the concrete transformer $\mathscr{F}^*[\![\mathsf{S}]\!]$ so its traces must be of the form $\pi\langle \mathsf{at}[\![\mathsf{S}]\!],\, \rho\rangle$⟆

$= \{\langle \pi_2\langle \mathsf{at}[\![\mathsf{S}]\!],\, \rho\rangle\langle \mathsf{aft}[\![\mathsf{S}]\!],\, \rho\rangle,\, \mathsf{R}'\rangle \mid \langle \pi_2\langle \mathsf{at}[\![\mathsf{S}]\!],\, \rho\rangle,\, \mathsf{R}''\rangle \in \mathscr{M}^\natural\langle \underline{\varrho},\, \mathsf{R}\rangle X \wedge \mathscr{B}[\![\mathsf{B}]\!]\, \rho = \mathsf{ff} \wedge \mathscr{M}^t\langle \underline{\varrho},\, \mathsf{R}''\rangle(\langle \mathsf{at}[\![\mathsf{S}]\!],\, \rho\rangle\langle \mathsf{aft}[\![\mathsf{S}]\!],\, \rho\rangle) = \langle \mathsf{tt},\, \mathsf{R}'\rangle\}$ ⟅(19) ⟆

$= \{\langle \pi_2\langle \mathsf{at}[\![\mathsf{S}]\!],\, \rho\rangle\langle \mathsf{aft}[\![\mathsf{S}]\!],\, \rho\rangle,\, \varepsilon\rangle \mid \langle \pi_2\langle \mathsf{at}[\![\mathsf{S}]\!],\, \rho\rangle,\, \varepsilon\rangle \in \mathscr{M}^\natural\langle \underline{\varrho},\, \mathsf{R}\rangle X \wedge \mathscr{B}[\![\mathsf{B}]\!]\, \rho = \mathsf{ff}\} \cup \{\langle \pi_2\langle \mathsf{at}[\![\mathsf{S}]\!],\, \rho\rangle\langle \mathsf{aft}[\![\mathsf{S}]\!],\, \rho\rangle,\, \mathsf{R}'\rangle \mid \langle \pi_2\langle \mathsf{at}[\![\mathsf{S}]\!],\, \rho\rangle,\, \mathsf{R}''\rangle \in \mathscr{M}^\natural\langle \underline{\varrho},\, \mathsf{R}\rangle X \wedge \mathscr{B}[\![\mathsf{B}]\!]\, \rho = \mathsf{ff} \wedge \mathsf{R}'' \notin \mathscr{R}_\varepsilon \wedge \mathscr{M}^t\langle \underline{\varrho},\, \mathsf{R}''\rangle(\langle \mathsf{at}[\![\mathsf{S}]\!],\, \rho\rangle\langle \mathsf{aft}[\![\mathsf{S}]\!],\, \rho\rangle) = \langle \mathsf{tt},\, \mathsf{R}'\rangle\}$

⟅case analysis and $\mathscr{M}^t\langle \underline{\varrho},\, \varepsilon\rangle\pi \triangleq \langle \mathsf{tt},\, \varepsilon\rangle$ in (18)⟆

$= \{\langle \pi_2\langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle\langle \mathsf{aft}[\![\mathtt{S}]\!],\ \rho\rangle,\ \varepsilon\rangle \mid \langle \pi_2\langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle,\ \varepsilon\rangle \in \boldsymbol{\mathscr{M}}^{\natural}\langle \underline{\varrho},\ \mathtt{R}\rangle X \wedge \boldsymbol{\mathscr{B}}[\![\mathtt{B}]\!]\,\rho = \mathtt{ff}\} \cup$
$\{\langle \pi_2\langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle\langle \mathsf{aft}[\![\mathtt{S}]\!],\ \rho\rangle,\ \varepsilon\rangle \mid \langle \pi_2\langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle,\ \mathtt{R}''\rangle \in \boldsymbol{\mathscr{M}}^{\natural}\langle \underline{\varrho},\ \mathtt{R}\rangle X \wedge \boldsymbol{\mathscr{B}}[\![\mathtt{B}]\!]\rho = \mathtt{ff} \wedge \mathtt{R}'' \notin \mathit{R}_{\varepsilon} \wedge \langle \mathtt{L}'\ \mathtt{:}\ \mathtt{B}',\ \mathtt{R}'\rangle = \mathsf{fstnxt}(\mathtt{R}'') \wedge \mathtt{R}' \in \mathit{R}_{\varepsilon} \wedge \langle \underline{\varrho},\ \langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle\rangle \in \boldsymbol{\mathscr{S}}^{r}[\![\mathtt{L}'\ \mathtt{:}\ \mathtt{B}']\!]\} \cup$
$\{\langle \pi_2\langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle\langle \mathsf{aft}[\![\mathtt{S}]\!],\ \rho\rangle,\ \mathtt{R}'\rangle \mid \langle \pi_2\langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle,\ \mathtt{R}''\rangle \in \boldsymbol{\mathscr{M}}^{\natural}\langle \underline{\varrho},\ \mathtt{R}\rangle X \wedge \boldsymbol{\mathscr{B}}[\![\mathtt{B}]\!]\ \rho = \mathtt{ff} \wedge \mathtt{R}'' \notin \mathit{R}_{\varepsilon} \wedge \langle \mathtt{L}'\ \mathtt{:}\ \mathtt{B}',\ \mathtt{R}'''\rangle = \mathsf{fstnxt}(\mathtt{R}'') \wedge \langle \underline{\varrho},\ \langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle\rangle \in \boldsymbol{\mathscr{S}}^{r}[\![\mathtt{L}'\ \mathtt{:}\ \mathtt{B}']\!] \wedge \mathtt{R}''' \notin \mathit{R}_{\varepsilon} \wedge \langle \mathtt{L}''\ \mathtt{:}\ \mathtt{B}'',\ \mathtt{R}'\rangle = \mathsf{fstnxt}(\mathtt{R}''') \wedge \langle \underline{\varrho},\ \langle \mathsf{aft}[\![\mathtt{S}]\!],\ \rho\rangle\rangle \in \boldsymbol{\mathscr{S}}^{r}[\![\mathtt{L}''\ \mathtt{:}\ \mathtt{B}'']\!]\}$

(since $(\langle \mathtt{tt},\ \mathtt{R}'\rangle = \boldsymbol{\mathscr{M}}^{t}\langle \underline{\varrho},\ \mathtt{R}''\rangle(\langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle\langle \mathsf{aft}[\![\mathtt{S}]\!],\ \rho\rangle)) \Leftrightarrow (\langle \mathtt{L}'\ \mathtt{:}\ \mathtt{B}',\ \mathtt{R}'\rangle = \mathsf{fstnxt}(\mathtt{R}'') \wedge \mathtt{R}' \in \mathit{R}_{\varepsilon} \wedge \langle \underline{\varrho},\ \langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle\rangle \in \boldsymbol{\mathscr{S}}^{r}[\![\mathtt{L}'\ \mathtt{:}\ \mathtt{B}']\!]) \vee (\langle \mathtt{L}'\ \mathtt{:}\ \mathtt{B}',\ \mathtt{R}'''\rangle = \mathsf{fstnxt}(\mathtt{R}'') \wedge \langle \underline{\varrho},\ \langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle\rangle \in \boldsymbol{\mathscr{S}}^{r}[\![\mathtt{L}'\ \mathtt{:}\ \mathtt{B}']\!] \wedge \mathtt{R}''' \notin \mathit{R}_{\varepsilon} \wedge \langle \mathtt{L}''\ \mathtt{:}\ \mathtt{B}'',\ \mathtt{R}'\rangle = \mathsf{fstnxt}(\mathtt{R}''') \wedge \langle \underline{\varrho},\ \langle \mathsf{aft}[\![\mathtt{S}]\!],\ \rho\rangle\rangle \in \boldsymbol{\mathscr{S}}^{r}[\![\mathtt{L}''\ \mathtt{:}\ \mathtt{B}'']\!])$ as shown above while proving the second term in case (31) of a conditional statement $\mathtt{S} ::= \mathtt{if}^{\,\ell}\ \mathtt{(B)}\ \mathtt{S}_t$)

— The third and last case is that of an iteration executing the loop body.

$\boldsymbol{\mathscr{M}}^{\natural}\langle \underline{\varrho},\ \mathtt{R}\rangle(\{\pi_2\langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle\langle \mathsf{at}[\![\mathtt{S}_b]\!],\ \rho\rangle \cdot \pi_3 \mid \pi_2\langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle \in X \wedge \boldsymbol{\mathscr{B}}[\![\mathtt{B}]\!]\ \rho = \mathtt{tt} \wedge \langle \mathsf{at}[\![\mathtt{S}_b]\!],\ \rho\rangle\pi_3 \in \boldsymbol{\mathscr{S}}^{*}[\![\mathtt{S}_b]\!]\})$

$= \{\langle \pi, \mathtt{R}'\rangle \mid \pi \in \{\pi_2\langle \mathsf{at}[\![\mathtt{S}]\!], \rho\rangle\langle \mathsf{at}[\![\mathtt{S}_b]\!], \rho\rangle\pi_3 \mid \pi_2\langle \mathsf{at}[\![\mathtt{S}]\!], \rho\rangle \in X \wedge \boldsymbol{\mathscr{B}}[\![\mathtt{B}]\!]\rho = \mathtt{tt} \wedge \langle \mathsf{at}[\![\mathtt{S}_b]\!],\ \rho\rangle\pi_3 \in \boldsymbol{\mathscr{S}}^{*}[\![\mathtt{S}_b]\!]\} \wedge \langle \mathtt{tt},\ \mathtt{R}'\rangle = \boldsymbol{\mathscr{M}}^{t}\langle \underline{\varrho},\ \mathtt{R}\rangle\pi\}$ (19)

$= \{\langle \pi_2\langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle\langle \mathsf{at}[\![\mathtt{S}_b]\!],\ \rho\rangle\pi_3,\ \mathtt{R}'\rangle \mid \pi_2\langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle \in X \wedge \boldsymbol{\mathscr{B}}[\![\mathtt{B}]\!]\ \rho = \mathtt{tt} \wedge \langle \mathsf{at}[\![\mathtt{S}_b]\!],\ \rho\rangle\pi_3 \in \boldsymbol{\mathscr{S}}^{*}[\![\mathtt{S}_b]\!] \wedge \langle \mathtt{tt},\ \mathtt{R}'\rangle = \boldsymbol{\mathscr{M}}^{t}\langle \underline{\varrho},\ \mathtt{R}\rangle(\pi_2\langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle\langle \mathsf{at}[\![\mathtt{S}_b]\!],\ \rho\rangle\pi_3)\}$ (def. $\in$)

$= \{\langle \pi_2\langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle\langle \mathsf{at}[\![\mathtt{S}_b]\!],\ \rho\rangle\pi_3,\ \mathtt{R}'\rangle \mid \pi_2\langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle \in X \wedge \boldsymbol{\mathscr{B}}[\![\mathtt{B}]\!]\ \rho = \mathtt{tt} \wedge \langle \mathsf{at}[\![\mathtt{S}_b]\!],\ \rho\rangle\pi_3 \in \boldsymbol{\mathscr{S}}^{*}[\![\mathtt{S}_b]\!] \wedge \exists \mathtt{R}'' \in \mathit{R}\ .\ \boldsymbol{\mathscr{M}}^{t}\langle \underline{\varrho}, \mathtt{R}\rangle(\pi_2\langle \mathsf{at}[\![\mathtt{S}]\!], \rho\rangle) = \langle \mathtt{tt}, \mathtt{R}''\rangle \wedge \boldsymbol{\mathscr{M}}^{t}\langle \underline{\varrho},\ \mathtt{R}''\rangle(\langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle\langle \mathsf{at}[\![\mathtt{S}_b]\!],\ \rho\rangle\pi_3) = \langle \mathtt{tt},\ \mathtt{R}'\rangle\}$ (Lem. 18)

$= \{\langle \pi_2\langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle\langle \mathsf{at}[\![\mathtt{S}_b]\!],\ \rho\rangle\pi_3,\ \mathtt{R}'\rangle \mid \langle \pi_2\langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle,\ \mathtt{R}''\rangle \in \{\langle \pi,\ \mathtt{R}''\rangle \mid \pi \in X \wedge \langle \mathtt{tt},\ \mathtt{R}''\rangle = \boldsymbol{\mathscr{M}}^{t}\langle \underline{\varrho},\ \mathtt{R}\rangle\pi\} \wedge \boldsymbol{\mathscr{B}}[\![\mathtt{B}]\!]\ \rho = \mathtt{tt} \wedge \langle \mathsf{at}[\![\mathtt{S}_b]\!],\ \rho\rangle\pi_3 \in \boldsymbol{\mathscr{S}}^{*}[\![\mathtt{S}_b]\!] \wedge \boldsymbol{\mathscr{M}}^{t}\langle \underline{\varrho},\ \mathtt{R}''\rangle(\langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle\langle \mathsf{at}[\![\mathtt{S}_b]\!],\ \rho\rangle\pi_3) = \langle \mathtt{tt},\ \mathtt{R}'\rangle\}$
(def. $\in$ and X is an iterate of the concrete transformer $\boldsymbol{\mathscr{F}}^{*}[\![\mathtt{S}]\!]$ so its traces must be of the form $\pi_2\langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle$)

$= \{\langle \pi_2\langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle\langle \mathsf{at}[\![\mathtt{S}_b]\!],\ \rho\rangle\pi_3,\ \mathtt{R}'\rangle \mid \langle \pi_2\langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle,\ \mathtt{R}''\rangle \in \boldsymbol{\mathscr{M}}^{\natural}\langle \underline{\varrho},\ \mathtt{R}\rangle X \wedge \boldsymbol{\mathscr{B}}[\![\mathtt{B}]\!]\ \rho = \mathtt{tt} \wedge \langle \mathsf{at}[\![\mathtt{S}_b]\!],\ \rho\rangle\pi_3 \in \boldsymbol{\mathscr{S}}^{*}[\![\mathtt{S}_b]\!] \wedge \boldsymbol{\mathscr{M}}^{t}\langle \underline{\varrho},\ \mathtt{R}''\rangle(\langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle\langle \mathsf{at}[\![\mathtt{S}_b]\!],\ \rho\rangle\pi_3) = \langle \mathtt{tt},\ \mathtt{R}'\rangle\}$

(19)

$= \{\langle \pi_2\langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle\langle \mathsf{at}[\![\mathtt{S}_b]\!],\ \rho\rangle\pi_3,\ \mathtt{R}'\rangle \mid \langle \pi_2\langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle,\ \mathtt{R}''\rangle \in \boldsymbol{\mathscr{M}}^{\natural}\langle \underline{\varrho},\ \mathtt{R}\rangle X \wedge \boldsymbol{\mathscr{B}}[\![\mathtt{B}]\!]\ \rho = \mathtt{tt} \wedge \langle \mathsf{at}[\![\mathtt{S}_b]\!],\ \rho\rangle\pi_3 \in \boldsymbol{\mathscr{S}}^{*}[\![\mathtt{S}_b]\!] \wedge (\exists \mathtt{R}''' \in \mathit{R}\ .\ \boldsymbol{\mathscr{M}}^{t}\langle \underline{\varrho},\ \mathtt{R}''\rangle(\langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle\langle \mathsf{at}[\![\mathtt{S}_b]\!],\ \rho\rangle) = \langle \mathtt{tt},\ \mathtt{R}'''\rangle \wedge \boldsymbol{\mathscr{M}}^{t}\langle \underline{\varrho},\ \mathtt{R}'''\rangle(\langle \mathsf{at}[\![\mathtt{S}_b]\!],\ \rho\rangle\pi_3) = \langle \mathtt{tt},\ \mathtt{R}'\rangle)\}$ (Lem. 18)

$= \{\langle \pi_2\langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle\langle \mathsf{at}[\![\mathtt{S}_b]\!],\ \rho\rangle\pi_3,\ \mathtt{R}'\rangle \mid \langle \pi_2\langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle,\ \mathtt{R}''\rangle \in \boldsymbol{\mathscr{M}}^{\natural}\langle \underline{\varrho},\ \mathtt{R}\rangle X \wedge \boldsymbol{\mathscr{B}}[\![\mathtt{B}]\!]\ \rho = \mathtt{tt} \wedge \exists \mathtt{R}''' \in \mathit{R}\ .\ \langle \langle \mathsf{at}[\![\mathtt{S}_b]\!],\ \rho\rangle\pi_3,\ \mathtt{R}'\rangle \in \{\langle \pi,\ \mathtt{R}'\rangle \mid \pi \in \boldsymbol{\mathscr{S}}^{*}[\![\mathtt{S}_b]\!] \wedge \langle \mathtt{tt},\ \mathtt{R}'\rangle = \boldsymbol{\mathscr{M}}^{t}\langle \underline{\varrho},\ \mathtt{R}'''\rangle\pi\} \wedge \boldsymbol{\mathscr{M}}^{t}\langle \underline{\varrho},\ \mathtt{R}''\rangle(\langle \mathsf{at}[\![\mathtt{S}]\!],\ \rho\rangle\langle \mathsf{at}[\![\mathtt{S}_b]\!],\ \rho\rangle) = \langle \mathtt{tt},\ \mathtt{R}'''\rangle\}$

⟅def. ∈ and def. $\mathcal{S}^*[\![\mathsf{S}_b]\!]$ in Section **2** so that its traces must be of the form $\langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle\pi_3$⟆

$= \{\langle \pi_2\langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle\langle \mathsf{at}[\![\mathsf{S}_b]\!],\ \rho\rangle\pi_3,\ \mathsf{R}'\rangle \mid \langle \pi_2\langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle,\ \mathsf{R}''\rangle \in \mathcal{M}^{\downarrow}\langle \underline{\varrho},\ \mathsf{R}\rangle X \wedge \mathcal{B}[\![\mathsf{B}]\!]\ \rho =$
$\mathsf{tt} \wedge \mathcal{M}^t\langle \underline{\varrho}, \mathsf{R}''\rangle(\langle \mathsf{at}[\![\mathsf{S}]\!], \rho\rangle\langle \mathsf{at}[\![\mathsf{S}_b]\!], \rho\rangle) = \langle \mathsf{tt}, \mathsf{R}'''\rangle \wedge \langle\langle \mathsf{at}[\![\mathsf{S}_b]\!], \rho\rangle\pi_3, \mathsf{R}'\rangle \in \mathcal{M}^{\downarrow}[\![\mathsf{S}_b]\!]\langle \underline{\varrho},$
$\mathsf{R}'''\rangle\}$ ⟅(20) and (19), $\wedge$ commutative⟆

There are two subcases depending on whether $\mathsf{R}'' \in \mathcal{R}_\varepsilon$ or not.

— If $\mathsf{R}'' \in \mathcal{R}_\varepsilon$, then

$= \{\langle \pi_2\langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle\langle \mathsf{at}[\![\mathsf{S}_b]\!],\ \rho\rangle\pi_3,\ \varepsilon\rangle \mid \langle \pi_2\langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle,\ \varepsilon\rangle \in \mathcal{M}^{\downarrow}\langle \underline{\varrho},\ \mathsf{R}\rangle X \wedge \mathcal{B}[\![\mathsf{B}]\!]\ \rho =$
$\mathsf{tt} \wedge \langle \mathsf{at}[\![\mathsf{S}_b]\!],\ \rho\rangle\pi_3 \in \mathcal{S}^*[\![\mathsf{S}_b]\!]\}$
⟅since $\mathsf{R}'' \in \mathcal{R}_\varepsilon$ and $\mathcal{M}^t\langle \underline{\varrho},\ \mathsf{R}''\rangle(\langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle\langle \mathsf{at}[\![\mathsf{S}_b]\!],\ \rho\rangle) = \langle \mathsf{tt},\ \mathsf{R}'''\rangle$ imply that $\mathsf{R}''' = \varepsilon$ by (18) and so $\langle\langle \mathsf{at}[\![\mathsf{S}_b]\!],\ \rho\rangle\pi_3,\ \mathsf{R}'\rangle \in \mathcal{M}^{\downarrow}[\![\mathsf{S}_b]\!]\langle \underline{\varrho},\ \mathsf{R}'''\rangle = \{\langle \pi,\ \varepsilon\rangle \mid \pi \in \mathcal{S}^*[\![\mathsf{S}_b]\!]\}$ by (20) and (19) implies $\mathsf{R}' = \varepsilon$ and $\langle \mathsf{at}[\![\mathsf{S}_b]\!],\ \rho\rangle\pi_3 \in \mathcal{S}^*[\![\mathsf{S}_b]\!]$⟆

— Otherwise $\mathsf{R}'' \notin \mathcal{R}_\varepsilon$

$= \{\langle \pi_2\langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle\langle \mathsf{at}[\![\mathsf{S}_b]\!],\ \rho\rangle\pi_3,\ \mathsf{R}'\rangle \mid \langle \pi_2\langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle,\ \mathsf{R}''\rangle \in \mathcal{M}^{\downarrow}\langle \underline{\varrho},\ \mathsf{R}\rangle X \wedge \mathcal{B}[\![\mathsf{B}]\!]\ \rho =$
$\mathsf{tt} \wedge \mathsf{R}'' \notin \mathcal{R}_\varepsilon \wedge \langle \mathsf{L} : \mathsf{B},\ \mathsf{R}''''\rangle = \mathsf{fstnxt}(\mathsf{R}'') \wedge \langle \underline{\varrho},\ \langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle\rangle \in \mathcal{S}^r[\![\mathsf{L} : \mathsf{B}]\!] \wedge \mathcal{M}^t\langle \underline{\varrho},$
$\mathsf{R}''''\rangle\langle \mathsf{at}[\![\mathsf{S}_b]\!],\ \rho\rangle = \langle \mathsf{tt},\ \mathsf{R}'''\rangle \wedge \langle\langle \mathsf{at}[\![\mathsf{S}_b]\!],\ \rho\rangle\pi_3,\ \mathsf{R}'\rangle \in \mathcal{M}^{\downarrow}[\![\mathsf{S}_b]\!]\langle \underline{\varrho},\ \mathsf{R}'''\rangle\}$ ⟅(18)⟆

There are two subsubcases, depending on whether R'''' is empty or not.

– If $\mathsf{R}'''' \in \mathcal{R}_\varepsilon$ then, as shown before, $\mathcal{M}^t\langle \underline{\varrho},\ \mathsf{R}''''\rangle\langle \mathsf{at}[\![\mathsf{S}_b]\!],\ \rho\rangle = \langle \mathsf{tt},\ \mathsf{R}'''\rangle$ implies that $\mathsf{R}''' \in \mathcal{R}_\varepsilon$ and so $\langle\langle \mathsf{at}[\![\mathsf{S}_b]\!],\ \rho\rangle\pi_3,\ \mathsf{R}'\rangle \in \mathcal{M}^{\downarrow}[\![\mathsf{S}_b]\!]\langle \underline{\varrho},\ \mathsf{R}'''\rangle$ if and only if $\mathsf{R}' \in \mathcal{R}_\varepsilon$ and $\langle \mathsf{at}[\![\mathsf{S}_b]\!],\ \rho\rangle\pi_3 \in \mathcal{S}^*[\![\mathsf{S}_b]\!]$. We get

$= \{\langle \pi_2\langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle\langle \mathsf{at}[\![\mathsf{S}_b]\!],\ \rho\rangle\pi_3,\ \varepsilon\rangle \mid \langle \pi_2\langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle,\ \mathsf{R}''\rangle \in \mathcal{M}^{\downarrow}\langle \underline{\varrho},\ \mathsf{R}\rangle X \wedge \mathcal{B}[\![\mathsf{B}]\!]\ \rho =$
$\mathsf{tt} \wedge \mathsf{R}'' \notin \mathcal{R}_\varepsilon \wedge \langle \mathsf{L} : \mathsf{B}, \varepsilon\rangle = \mathsf{fstnxt}(\mathsf{R}'') \wedge \langle \underline{\varrho}, \langle \mathsf{at}[\![\mathsf{S}]\!], \rho\rangle\rangle \in \mathcal{S}^r[\![\mathsf{L} : \mathsf{B}]\!] \wedge \langle \mathsf{at}[\![\mathsf{S}_b]\!],\ \rho\rangle\pi_3 \in$
$\mathcal{S}^*[\![\mathsf{S}_b]\!]\}$ ⟅(18)⟆

– Otherwise $\mathsf{R}'''' \notin \mathcal{R}_\varepsilon$.

$= \{\langle \pi_2\langle \mathsf{at}[\![\mathsf{S}]\!], \rho\rangle\langle \mathsf{at}[\![\mathsf{S}_b]\!], \rho\rangle\pi_3, \mathsf{R}'\rangle \mid \langle \pi_2\langle \mathsf{at}[\![\mathsf{S}]\!], \rho\rangle, \mathsf{R}''\rangle \in \mathcal{M}^{\downarrow}\langle \underline{\varrho}, \mathsf{R}\rangle X \wedge \mathcal{B}[\![\mathsf{B}]\!]\rho = \mathsf{tt} \wedge$
$\mathsf{R}'' \notin \mathcal{R}_\varepsilon \wedge \langle \mathsf{L} : \mathsf{B}, \mathsf{R}''''\rangle = \mathsf{fstnxt}(\mathsf{R}'') \wedge \langle \underline{\varrho}, \langle \mathsf{at}[\![\mathsf{S}]\!], \rho\rangle\rangle \in \mathcal{S}^r[\![\mathsf{L} : \mathsf{B}]\!] \wedge \mathsf{R}'''' \notin \mathcal{R}_\varepsilon \wedge \mathcal{M}^t\langle \underline{\varrho},$
$\mathsf{R}''''\rangle\langle \mathsf{at}[\![\mathsf{S}_b]\!],\ \rho\rangle = \langle \mathsf{tt},\ \mathsf{R}'''\rangle \wedge \langle\langle \mathsf{at}[\![\mathsf{S}_b]\!],\ \rho\rangle\pi_3,\ \mathsf{R}'\rangle \in \mathcal{M}^{\downarrow}[\![\mathsf{S}_b]\!]\langle \underline{\varrho},\ \mathsf{R}'''\rangle\}$

$= \{\langle \pi_2\langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle\langle \mathsf{at}[\![\mathsf{S}_b]\!],\ \rho\rangle\pi_3,\ \mathsf{R}'\rangle \mid \langle \pi_2\langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle,\ \mathsf{R}''\rangle \in \mathcal{M}^{\downarrow}\langle \underline{\varrho},\ \mathsf{R}\rangle X \wedge \mathcal{B}[\![\mathsf{B}]\!]\ \rho =$
$\mathsf{tt} \wedge \mathsf{R}'' \notin \mathcal{R}_\varepsilon \wedge \langle \mathsf{L} : \mathsf{B},\ \mathsf{R}''''\rangle = \mathsf{fstnxt}(\mathsf{R}'') \wedge \langle \underline{\varrho},\ \langle \mathsf{at}[\![\mathsf{S}]\!],\ \rho\rangle\rangle \in \mathcal{S}^r[\![\mathsf{L} : \mathsf{B}]\!] \wedge \mathsf{R}'''' \notin$
$\mathcal{R}_\varepsilon \wedge \langle \mathsf{L}' : \mathsf{B}',\ \mathsf{R}'''\rangle = \mathsf{fstnxt}(\mathsf{R}'''') \wedge \langle \underline{\varrho},\ \langle \mathsf{at}[\![\mathsf{S}_b]\!],\ \rho\rangle\rangle \in \mathcal{S}^r[\![\mathsf{L}' : \mathsf{B}']\!] \wedge \langle\langle \mathsf{at}[\![\mathsf{S}_b]\!],$
$\rho\rangle\pi_3,\ \mathsf{R}'\rangle \in \mathcal{M}^{\downarrow}[\![\mathsf{S}_b]\!]\langle \underline{\varrho},\ \mathsf{R}'''\rangle\}$ ⟅(18)⟆

— Grouping all cases together we get the term (34) defining $\widehat{\mathcal{F}}^{\downarrow}[\![\mathsf{S}]\!]\langle \underline{\varrho}, \mathsf{R}\rangle(\mathcal{M}^{\downarrow}\langle \underline{\varrho},$ $\mathsf{R}\rangle X)$ and so Lem. 19 and the commutation condition $\mathcal{M}^{\downarrow}\langle \underline{\varrho},\ \mathsf{R}\rangle(\mathcal{F}^*[\![\mathsf{S}]\!](X)) = \widehat{\mathcal{F}}^{\downarrow}[\![\mathsf{S}]\!]\langle \underline{\varrho},\ \mathsf{R}\rangle\ (\mathcal{M}^{\downarrow}\langle \underline{\varrho},\ \mathsf{R}\rangle(X))$ for the iterates X of $\mathcal{F}^*[\![\mathsf{S}]\!]$ yield $\widehat{\mathcal{M}}^{\downarrow}[\![\mathsf{S}]\!]\langle \underline{\varrho},\ \mathsf{R}\rangle \triangleq \mathsf{lfp}^{\subseteq}\ (\widehat{\mathcal{F}}^{\downarrow}[\![\mathsf{S}]\!]\langle \underline{\varrho},\ \mathsf{R}\rangle)$ that is (33).

— In case (32) of a break statement $\mathtt{S} ::= \ell\,\mathtt{break}\ ;$

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

$= \{\langle \pi,\ \mathtt{R}'\rangle \mid \pi \in \mathscr{S}^*[\![\mathtt{S}]\!] \wedge \langle \mathtt{tt},\ \mathtt{R}'\rangle = \mathscr{M}^t\langle \underline{\varrho},\ \mathtt{R}\rangle\pi\}$ ⟅(20) and (19)⟆

$= \{\langle \pi,\ \mathtt{R}'\rangle \mid \pi \in \{\langle \ell,\ \rho\rangle \mid \rho \in \mathbb{E}\mathtt{v}\} \cup \{\langle \ell,\ \rho\rangle\langle \mathsf{brk\text{-}to}[\![\mathtt{S}]\!],\ \rho\rangle \mid \rho \in \mathbb{E}\mathtt{v}\} \wedge \langle \mathtt{tt},\ \mathtt{R}'\rangle = \mathscr{M}^t\langle \underline{\varrho},\ \mathtt{R}\rangle\pi\}$ ⟅(3)⟆

$= \{\langle\langle \ell,\ \rho\rangle,\ \mathtt{R}''\rangle \mid \langle \mathtt{tt},\ \mathtt{R}''\rangle = \mathscr{M}^t\langle \underline{\varrho},\ \mathtt{R}\rangle\langle \ell,\ \rho\rangle\} \cup \{\langle\langle \ell,\ \rho\rangle\langle \mathsf{brk\text{-}to}[\![\mathtt{S}]\!],\ \rho\rangle,\ \mathtt{R}''\rangle \mid \langle \mathtt{tt},\ \mathtt{R}''\rangle = \mathscr{M}^t\langle \underline{\varrho},\ \mathtt{R}\rangle(\langle \ell,\ \rho\rangle\langle \mathsf{brk\text{-}to}[\![\mathtt{S}]\!],\ \rho\rangle)\}$ ⟅def. ∪ and ∈⟆

$= \mathsf{let}\,\langle \mathtt{L}\ :\ \mathtt{B},\ \mathtt{R}'\rangle = \mathsf{fstnxt}(\mathtt{R})\ \mathsf{in}\ \{\langle\langle \ell,\ \rho\rangle,\ \mathtt{R}'\rangle \mid \langle \underline{\varrho},\ \langle \ell,\ \rho\rangle\rangle \in \mathscr{S}^r[\![\mathtt{L}\ :\ \mathtt{B}]\!]\} \cup \{\langle\langle \ell,\ \rho\rangle\langle \mathsf{brk\text{-}to}[\![\mathtt{S}]\!],\ \rho\rangle,\ \varepsilon\rangle \mid \mathtt{R}' \in \mathscr{R}_\varepsilon \wedge \langle \underline{\varrho},\ \langle \ell,\ \rho\rangle\rangle \in \mathscr{S}^r[\![\mathtt{L}\ :\ \mathtt{B}]\!]\} \cup \{\langle\langle \ell,\ \rho\rangle\langle \mathsf{brk\text{-}to}[\![\mathtt{S}]\!],\ \rho\rangle,\ \mathtt{R}''\rangle \mid \mathtt{R}' \notin \mathscr{R}_\varepsilon \wedge \langle \underline{\varrho},\ \langle \ell,\ \rho\rangle\rangle \in \mathscr{S}^r[\![\mathtt{L}\ :\ \mathtt{B}]\!] \wedge \langle \mathtt{L}'\ :\ \mathtt{B}',\ \mathtt{R}''\rangle = \mathsf{fstnxt}(\mathtt{R}') \wedge \langle \underline{\varrho},\ \langle \mathsf{brk\text{-}to}[\![\mathtt{S}]\!],\ \rho\rangle\rangle \in \mathscr{S}^r[\![\mathtt{L}'\ :\ \mathtt{B}']\!]\}$ ⟅$\mathtt{R} \notin \mathscr{R}_\varepsilon$, case analysis on $\mathtt{R}' \in \mathscr{R}_\varepsilon$, and(18)⟆

— The proofs are similar in the cases (36) of an alternative statement $\mathtt{S} ::= \mathtt{if}\ \ell\ \mathtt{(B)}\ \mathtt{S}_t\ \mathtt{else}\ \mathtt{S}_f$ and (37) of a a compound statement $\mathtt{S} ::= \mathtt{\{\ Sl\ \}}$ □

9. Notes on implementations and expressivity

Of course further hypotheses and refinements would be necessary to get an effective algorithm as specified by the Def. 17 of structural model checking. A common hypothesis in model checking is that the set of states $\mathbb{S}$ is finite. Traces may still be infinite so the fixpoint computation (33) may not converge. However, infinite traces on finite states must involve an initial finite prefix followed by a finite cycle (often called a lasso) [34]. It follows that the infinite set of prefix traces can be finitely represented by a finite set of maximal finite traces and finite lassos. Regular expressions $\mathtt{L}\ :\ \mathtt{B}$ can be attached to the states as determined by the analysis, and there are finitely many of them in the specification. These finiteness properties can be taken into account to ensure the convergence of the fixpoint computation in (33).

A symbolic representation of the states in finite/lasso traces may be useful as in symbolic execution [32] or using BDDs [6] for boolean encodings of programs. By Kleene theorem [45, Theorem 2.1, p. 87], a convenient representation of regular expressions is by (deterministic) finite automata *e.g.* [37]. Symbolic automata-based algorithms can be used to implement a data structure for operations over sets of sequences [26].

Of course the hypothesis that the state space is finite and small enough to scale up and limit the combinatorial blow up of the finite state-space is unrealistic [11]. In practice, the set of states $\mathbb{S}$ is very large, so abstraction and a widening/dual narrowing are necessary. A typical trivial widening is bounded model checking (*e.g.* widen to all states after n fixpoint iterations) [5]. Those of [39] for BDDs are more elaborated.

10. Conclusion

We have illustrated the idea that model checking is an abstract interpretation, as first introduced in [17]. This point of view also yields specification-

preserving abstract model checking [18] as well as abstraction refinement algorithms [23].

Specifications by temporal logics are not commonly accepted by programmers. For example, in [40], the specifications had to be written by academics. Regular expressions or path expressions [8] or more expressive extensions such as [25, 49, 20, 7, 30, 22] might turn out to be more familiar. Moreover, for security monitors the false alarms of the static analysis can be checked at runtime [46, 38].

Convergence of model checking requires expressivity restrictions on both the considered models of computation and the considered temporal logics. For some expressive models of computation and temporal logics, state finiteness is not enough to guarantee termination of model checking [17, 24]. Finite enumeration is limited, even with symbolic encodings. Beyond finiteness, scalability is always a problem with model checking and the regular software model checking algorithm $\widehat{\mathscr{M}}$ is no exception, so abstraction and induction are ultimately required to reason on programs.

Most often, abstract model checking uses homomorphic/partitioning abstractions *e.g.* [4]. This is because the abstraction of a transition system on concrete states is a transition system on abstract states so model checkers are reusable in the abstract. However, excluding edgy abstractions as in [13], state-based finite abstraction is very restrictive [24] and do not guarantee scalability (*e.g.* SLAM [3]). Such restrictions on abstractions do not apply to structural model checking so that abstractions more powerful than partitioning can be considered.

As an alternative approach, a regular expression can be automatically extracted by static analysis of the program trace semantics that recognizes all feasible execution paths and usually more [19]. Then model-checking a regular specification becomes a regular language inclusion problem [36].

Acknowledgements

Research funded in part by NSF Grant CCF-1617717.

Appendix A. Syntactic component relation

The strict syntactic order $\lhd$ for the programs of Section **2.1** is defined as follows.

$$
\begin{array}{rcll}
\mathtt{P} & ::= & \mathtt{Sl}\ \ell & \mathtt{Sl} \lhd \mathtt{P} \\
\mathtt{S} & ::= & & \\
 & & \mathtt{x = E\ ;} & \mathtt{x} \lhd \mathtt{S}, \quad \mathtt{E} \lhd \mathtt{S} \\
 & | & \mathtt{;} & \\
 & | & \mathtt{if\ (B)}\ \mathtt{S}_t & \mathtt{B} \lhd \mathtt{S}, \quad \mathtt{S}_t \lhd \mathtt{S} \\
 & | & \mathtt{if\ (B)}\ \mathtt{S}_t\ \mathtt{else}\ \mathtt{S}_f & \mathtt{B} \lhd \mathtt{S}, \quad \mathtt{S}_t \lhd \mathtt{S}, \quad \mathtt{S}_f \lhd \mathtt{S} \\
 & | & \mathtt{while\ (B)}\ \mathtt{S}_b & \mathtt{B} \lhd \mathtt{S}, \quad \mathtt{S}_b \lhd \mathtt{S} \\
 & | & \mathtt{break\ ;} & \\
 & | & \mathtt{\{\ Sl\ \}} & \mathtt{Sl} \lhd \mathtt{S} \\
\mathtt{Sl} & ::= & \mathtt{Sl'\ S} \mid \epsilon & \mathtt{Sl'} \lhd \mathtt{Sl}, \quad \mathtt{S} \lhd \mathtt{Sl}, \quad \epsilon \lhd \mathtt{Sl}
\end{array}
$$

$\lhd$ is well-founded since it is defined on program components which have finite tree representations and $\mathtt{S'} \lhd \mathtt{S}$ implies that finite tree representation of $\mathtt{S'}$ has at least one node less than the tree representation of $\mathtt{S}$.

Appendix B. Program labeling

The program labeling informally introduced in Section **2.2** is left unspecified but must satisfies the following postulates. $\mathsf{at}[\![\mathtt{S}]\!]$ is the program point at which execution of program component $\mathtt{S}$ starts. $\mathsf{aft}[\![\mathtt{S}]\!]$ is the program point at which execution of $\mathtt{S}$ is supposed to terminate, if ever. After executing program component $\mathtt{S}$, execution will continue at that program point (unless there is a **break ;** or the statement does not terminate).

$$
\begin{array}{rcllll}
\mathtt{P} & ::= & \mathtt{Sl} & \mathsf{at}[\![\mathtt{P}]\!] \triangleq \mathsf{at}[\![\mathtt{Sl}]\!] & \mathsf{aft}[\![\mathtt{P}]\!] \triangleq \mathsf{aft}[\![\mathtt{Sl}]\!] & \\
\mathtt{Sl} & ::= & \mathtt{Sl'\ S} & \mathsf{at}[\![\mathtt{Sl}]\!] \triangleq \mathsf{at}[\![\mathtt{Sl'}]\!] & \mathsf{aft}[\![\mathtt{Sl'}]\!] \triangleq \mathsf{at}[\![\mathtt{S}]\!], & \mathsf{aft}[\![\mathtt{S}]\!] \triangleq \mathsf{aft}[\![\mathtt{Sl}]\!] \\
\mathtt{Sl} & ::= & \epsilon & \mathsf{at}[\![\mathtt{Sl}]\!] \triangleq \mathsf{aft}[\![\mathtt{Sl}]\!] & & \\
\mathtt{S} & ::= & \mathtt{\{\ Sl\ \}} & \mathsf{at}[\![\mathtt{S}]\!] \triangleq \mathsf{at}[\![\mathtt{Sl}]\!] & & \\
\mathtt{S} & ::= & \mathtt{if\ (B)}\ \mathtt{S}_t & & \mathsf{aft}[\![\mathtt{S}_t]\!] \triangleq \mathsf{aft}[\![\mathtt{S}]\!] & \\
\mathtt{S} & ::= & \mathtt{if\ (B)}\ \mathtt{S}_t\ \mathtt{else}\ \mathtt{S}_f & & \mathsf{aft}[\![\mathtt{S}_t]\!] \triangleq \mathsf{aft}[\![\mathtt{S}_f]\!] \triangleq \mathsf{aft}[\![\mathtt{S}]\!] & \\
\mathtt{S} & ::= & \mathtt{while\ (B)}\ \mathtt{S}_b & & \mathsf{aft}[\![\mathtt{S}_b]\!] \triangleq \mathsf{at}[\![\mathtt{S}]\!] & \\
\mathtt{S} & ::= & \mathtt{\{\ Sl\ \}} & \mathsf{at}[\![\mathtt{S}]\!] \triangleq \mathsf{at}[\![\mathtt{Sl}]\!] & \mathsf{aft}[\![\mathtt{Sl}]\!] \triangleq \mathsf{aft}[\![\mathtt{S}]\!] &
\end{array}
$$

We use explicit labelling in examples and proofs as the following shorthand for labels.

$$
\begin{array}{rcll}
\mathtt{S} & ::= & \ell\ \mathtt{x = E\ ;} & \ell = \mathsf{at}[\![\mathtt{S}]\!] \\
\mathtt{S} & ::= & \ell\mathtt{;} & \ell = \mathsf{at}[\![\mathtt{S}]\!] \\
\mathtt{S} & ::= & \mathtt{if}\ \ell\ \mathtt{(B)}\ \mathtt{S}_t & \ell = \mathsf{at}[\![\mathtt{S}]\!] \\
\mathtt{S} & ::= & \mathtt{if}\ \ell\ \mathtt{(B)}\ \mathtt{S}_t\ \mathtt{else}\ \mathtt{S}_f & \ell = \mathsf{at}[\![\mathtt{S}]\!] \\
\mathtt{S} & ::= & \mathtt{while}\ \ell\ \mathtt{(B)}\ \mathtt{S}_b & \ell = \mathsf{at}[\![\mathtt{S}]\!] \\
\mathtt{S} & ::= & \ell\ \mathtt{break\ ;} & \ell = \mathsf{at}[\![\mathtt{S}]\!] \\
\mathtt{P} & ::= & \mathtt{Sl}\ \ell & \ell = \mathsf{aft}[\![\mathtt{P}]\!] = \mathsf{aft}[\![\mathtt{Sl}]\!]
\end{array}
$$

$\mathsf{esc}[\![\mathtt{S}]\!]$ is $\mathtt{tt}$ if and only if the program component $\mathtt{S} \in \mathbb{P}\mathbb{c}$ contains a **`break ;`** statement escaping out of that component $\mathtt{S}$ to the program point $\mathsf{brk\text{-}to}[\![\mathtt{S}]\!]$ (which is well-defined only when $\mathsf{esc}[\![\mathtt{S}]\!] = \mathtt{tt}$).

`P`	::=	`Sl`	$\mathsf{esc}[\![\mathtt{P}]\!] \triangleq \mathsf{esc}[\![\mathtt{Sl}]\!], \quad \mathsf{esc}[\![\mathtt{P}]\!] = \mathtt{ff}$
`Sl`	::=	`Sl' S`	$\mathsf{esc}[\![\mathtt{Sl}]\!] \triangleq \mathsf{esc}[\![\mathtt{Sl'}]\!] \vee \mathsf{esc}[\![\mathtt{S}]\!]$
			$\mathsf{brk\text{-}to}[\![\mathtt{Sl'}]\!] \triangleq \mathsf{brk\text{-}to}[\![\mathtt{S}]\!] \triangleq \mathsf{brk\text{-}to}[\![\mathtt{Sl}]\!]$
`Sl`	::=	ϵ	$\mathsf{esc}[\![\mathtt{Sl}]\!] \triangleq \mathtt{ff}$
`S`	::=	`x = E ;`	$\mathsf{esc}[\![\mathtt{S}]\!] \triangleq \mathtt{ff}$
`S`	::=	`;`	$\mathsf{esc}[\![\mathtt{S}]\!] \triangleq \mathtt{ff}$
`S`	::=	`if (B)` $\mathtt{S}_t$	$\mathsf{esc}[\![\mathtt{S}]\!] \triangleq \mathsf{esc}[\![\mathtt{S}_t]\!], \quad \mathsf{brk\text{-}to}[\![\mathtt{S}_t]\!] \triangleq \mathsf{brk\text{-}to}[\![\mathtt{S}]\!]$
`S`	::=	`if (B)` $\mathtt{S}_t$ `else` $\mathtt{S}_f$	$\mathsf{esc}[\![\mathtt{S}]\!] \triangleq \mathsf{esc}[\![\mathtt{S}_t]\!] \vee \mathsf{esc}[\![\mathtt{S}_f]\!]$
			$\mathsf{brk\text{-}to}[\![\mathtt{S}_t]\!] \triangleq \mathsf{brk\text{-}to}[\![\mathtt{S}_f]\!] \triangleq \mathsf{brk\text{-}to}[\![\mathtt{S}]\!]$
`S`	::=	`while (B)` $\mathtt{S}_b$	$\mathsf{esc}[\![\mathtt{S}]\!] \triangleq \mathtt{ff}, \quad \mathsf{brk\text{-}to}[\![\mathtt{S}_b]\!] \triangleq \mathsf{aft}[\![\mathtt{S}]\!]$
`S`	::=	**`break ;`**	$\mathsf{esc}[\![\mathtt{S}]\!] \triangleq \mathtt{tt}$
`S`	::=	`{ Sl }`	$\mathsf{esc}[\![\mathtt{S}]\!] \triangleq \mathsf{esc}[\![\mathtt{Sl}]\!], \quad \mathsf{brk\text{-}to}[\![\mathtt{Sl}]\!] \triangleq \mathsf{brk\text{-}to}[\![\mathtt{S}]\!]$

$\mathsf{brks\text{-}of}[\![\mathtt{S}]\!]$ collects the labels of all **`break ;`** program components that can escape out of `S` (so excluding **`break ;`** statements inside an iteration statement within `S`). The definition checks that **`break ;`** statements can only appear within loops;

`P`	::=	`Sl`	$\mathsf{brks\text{-}of}[\![\mathtt{P}]\!] \triangleq \mathsf{brks\text{-}of}[\![\mathtt{Sl}]\!]$	$\mathsf{brks\text{-}of}[\![\mathtt{P}]\!] = \varnothing$
`Sl`	::=	`Sl' S`	$\mathsf{brks\text{-}of}[\![\mathtt{Sl}]\!] \triangleq \mathsf{brks\text{-}of}[\![\mathtt{Sl'}]\!] \cup \mathsf{brks\text{-}of}[\![\mathtt{S}]\!]$	
`Sl`	::=	ϵ	$\mathsf{brks\text{-}of}[\![\mathtt{Sl}]\!] \triangleq \varnothing$	
`S`	::=	`x = E ;`	$\mathsf{brks\text{-}of}[\![\mathtt{S}]\!] \triangleq \varnothing$	
`S`	::=	`;`	$\mathsf{brks\text{-}of}[\![\mathtt{S}]\!] \triangleq \varnothing$	
`S`	::=	`if (B)` $\mathtt{S}_t$	$\mathsf{brks\text{-}of}[\![\mathtt{S}]\!] \triangleq \mathsf{brks\text{-}of}[\![\mathtt{S}_t]\!]$	
`S`	::=	`if (B)` $\mathtt{S}_t$ `else` $\mathtt{S}_f$	$\mathsf{brks\text{-}of}[\![\mathtt{S}]\!] \triangleq \mathsf{brks\text{-}of}[\![\mathtt{S}_t]\!] \cup \mathsf{brks\text{-}of}[\![\mathtt{S}_f]\!]$	
`S`	::=	`while (B)` $\mathtt{S}_b$	$\mathsf{brks\text{-}of}[\![\mathtt{S}]\!] \triangleq \varnothing$	
`S`	::=	ℓ **`break ;`**	$\mathsf{brks\text{-}of}[\![\mathtt{S}]\!] \triangleq \{\mathsf{at}[\![\mathtt{S}]\!]\}$	
`S`	::=	`{ Sl }`	$\mathsf{brks\text{-}of}[\![\mathtt{S}]\!] \triangleq \mathsf{brks\text{-}of}[\![\mathtt{Sl}]\!]$	

$\mathsf{labs}[\![\mathtt{S}]\!]$ is the set of potentially reachable program points while executing the program component `S` either in or after `S`, or resulting from a break.

$$\mathsf{labs}[\![\mathtt{S}]\!] \quad \triangleq \quad \mathsf{in}[\![\mathtt{S}]\!] \cup \{\mathsf{aft}[\![\mathtt{S}]\!]\} \cup (\!|\, \mathsf{esc}[\![\mathtt{S}]\!] \;?\; \{\mathsf{brk\text{-}to}[\![\mathtt{S}]\!]\} \;:\; \varnothing \,|\!)$$

Proof (of Lem. 1). By structural induction on `S`.

In the base case, for example, if $\mathtt{Sl} ::= \epsilon$ then $\mathsf{in}[\![\mathtt{Sl}]\!] \triangleq \{\mathsf{at}[\![\mathtt{Sl}]\!]\}$ so $\mathsf{at}[\![\mathtt{Sl}]\!] \in \mathsf{in}[\![\mathtt{Sl}]\!]$ and for $\mathtt{S} ::= \ell$ `x = E ;` then $\mathsf{in}[\![\mathtt{S}]\!] \triangleq \{\ell\}$ where $\mathsf{at}[\![\mathtt{S}]\!] \triangleq \ell$. Similarly for the other base cases $\mathtt{S} ::=$ `;`, $\mathtt{S} ::=$ `if (B)` $\mathtt{S}_t$, $\mathtt{S} ::=$ `if (B)` $\mathtt{S}_t$ `else` $\mathtt{S}_f$, $\mathtt{S} ::=$ `while (B)` $\mathtt{S}_b$, and $\mathtt{S} ::=$ **`break ;`**.

For the induction cases, if $\mathtt{S} ::=$ `{ Sl }` then $\mathsf{at}[\![\mathtt{S}]\!] = \mathsf{at}[\![\mathtt{Sl}]\!] \in \mathsf{in}[\![\mathtt{Sl}]\!] = \mathsf{in}[\![\mathtt{S}]\!]$ by def. at, in, and induction hypothesis. If $\mathtt{Sl} ::= \mathtt{Sl'}\ \mathtt{S}$ then $\mathsf{at}[\![\mathtt{Sl}]\!] = \mathsf{at}[\![\mathtt{Sl'}]\!] \in$

$\mathsf{in}[\![\mathtt{Sl}']\!] \subseteq \mathsf{in}[\![\mathtt{Sl}]\!]$ by def. at, in, and induction hypothesis. Otherwise, $\mathtt{P} ::= \mathtt{Sl}\ \ell$ and $\mathsf{at}[\![\mathtt{P}]\!] = \mathsf{at}[\![\mathtt{Sl}]\!] \in \mathsf{in}[\![\mathtt{Sl}]\!] \subseteq \mathsf{in}[\![\mathtt{P}]\!]$ by def. at, in, and induction hypothesis. □

Proof (of Lem. 2). The proof is by induction on the distance $\delta(\mathtt{S})$ of $\mathtt{S}$ to the root of the abstract syntax tree of $\mathtt{P}$.

- For the basis $\mathtt{P} ::= \mathtt{Sl}\ \ell$, where $\delta(\mathtt{P}) = 0$, we have $\mathsf{aft}[\![\mathtt{P}]\!] \triangleq \mathsf{aft}[\![\mathtt{Sl}]\!] \triangleq \ell$ and $\mathsf{in}[\![\mathtt{P}]\!] \triangleq \mathsf{in}[\![\mathtt{Sl}]\!]$ with $\ell \notin \mathsf{in}[\![\mathtt{Sl}]\!]$ so $\mathsf{aft}[\![\mathtt{P}]\!] \notin \mathsf{in}[\![\mathtt{P}]\!]$ and $\mathsf{aft}[\![\mathtt{Sl}]\!] \notin \mathsf{in}[\![\mathtt{Sl}]\!]$.
- For $\mathtt{Sl} ::= \mathtt{Sl}'\ \mathtt{S}$ where $\delta(\mathtt{Sl}') = \delta(\mathtt{S}) = \delta(\mathtt{Sl}) + 1$, we have $\mathsf{aft}[\![\mathtt{Sl}']\!] \triangleq \mathsf{at}[\![\mathtt{S}]\!]$, $\mathsf{aft}[\![\mathtt{S}]\!] \triangleq \mathsf{aft}[\![\mathtt{Sl}]\!]$, $\mathsf{in}[\![\mathtt{Sl}]\!] \triangleq \mathsf{in}[\![\mathtt{Sl}']\!] \cup \mathsf{in}[\![\mathtt{S}]\!]$, $\mathsf{in}[\![\mathtt{Sl}']\!] \cap \mathsf{in}[\![\mathtt{S}]\!] = \varnothing$ since $\mathtt{Sl}' \neq \epsilon$ and, by Lem. 1, $\mathsf{at}[\![\mathtt{S}]\!] \in \mathsf{in}[\![\mathtt{S}]\!]$ so $\mathsf{aft}[\![\mathtt{Sl}']\!] = \mathsf{at}[\![\mathtt{S}]\!] \notin \mathsf{in}[\![\mathtt{Sl}']\!]$. Moreover, $\mathsf{aft}[\![\mathtt{S}]\!] = \mathsf{aft}[\![\mathtt{Sl}]\!] \notin \mathsf{in}[\![\mathtt{Sl}]\!]$ by induction hypothesis hence $\mathsf{aft}[\![\mathtt{S}]\!] \notin \mathsf{in}[\![\mathtt{S}]\!]$.
- If $\mathtt{S} ::= \mathtt{if}\ \ell\ \mathtt{(B)}\ \mathtt{S}_t$ then $\mathsf{aft}[\![\mathtt{S}_t]\!] \triangleq \mathsf{aft}[\![\mathtt{S}]\!]$, $\mathsf{aft}[\![\mathtt{S}]\!] \notin \mathsf{in}[\![\mathtt{S}]\!]$ by induction hypothesis since $\delta(\mathtt{S}_t) = \delta(\mathtt{S}) + 1$, so $\mathsf{aft}[\![\mathtt{S}_t]\!] \notin \mathsf{in}[\![\mathtt{S}_t]\!]$ since $\mathsf{in}[\![\mathtt{S}_t]\!] \subseteq \mathsf{in}[\![\mathtt{S}]\!]$.
- By a similar argument, $\mathsf{aft}[\![\mathtt{S}_t]\!] \notin \mathsf{in}[\![\mathtt{S}_t]\!]$ and $\mathsf{aft}[\![\mathtt{S}_f]\!] \notin \mathsf{in}[\![\mathtt{S}_f]\!]$ when $\mathtt{S} ::= \mathtt{if}\ \ell\ \mathtt{(B)}\ \mathtt{S}_t\ \mathtt{else}\ \mathtt{S}_f$.
- If $\mathtt{S} ::= \mathtt{while}\ \ell\ \mathtt{(B)}\ \mathtt{S}_b$ then $\mathsf{aft}[\![\mathtt{S}_b]\!] \triangleq \ell$ and $\ell \notin \mathsf{in}[\![\mathtt{S}_b]\!]$ by def. $\mathsf{in}[\![\mathtt{S}]\!]$.
- If $\mathtt{S} ::= \mathtt{\{\ Sl\ \}}$ and $\mathtt{Sl} \neq \mathtt{\{\ \ldots\{\ \epsilon\ \}\ldots\ \}}$ then $\mathsf{aft}[\![\mathtt{Sl}]\!] \triangleq \mathsf{aft}[\![\mathtt{S}]\!]$, $\mathsf{in}[\![\mathtt{S}]\!] \triangleq \mathsf{in}[\![\mathtt{Sl}]\!]$, and $\delta(\mathtt{Sl}) = \delta(\mathtt{S}) + 1$ so $\mathsf{aft}[\![\mathtt{S}]\!] \notin \mathsf{in}[\![\mathtt{S}]\!]$ by induction hypothesis since $\mathtt{S} \neq \epsilon$, proving $\mathsf{aft}[\![\mathtt{Sl}]\!] \notin \mathsf{in}[\![\mathtt{Sl}]\!]$. □

References

[1] Parosh Aziz Abdulla, Bengt Jonsson, Marcus Nilsson, and Mayank Saksena. A survey of regular model checking. In *CONCUR*, volume 3170 of *Lecture Notes in Computer Science*, pages 35–48. Springer, 2004.

[2] Rajeev Alur, Konstantinos Mamouras, and Dogan Ulus. Derivatives of quantitative regular expressions. In *Models, Algorithms, Logics and Tools*, volume 10460 of *Lecture Notes in Computer Science*, pages 75–95. Springer, 2017.

[3] Thomas Ball, Vladimir Levin, and Sriram K. Rajamani. A decade of software model checking with SLAM. *Commun. ACM*, 54(7):68–76, 2011.

[4] Thomas Ball, Andreas Podelski, and Sriram K. Rajamani. Boolean and cartesian abstraction for model checking C programs. In *TACAS*, volume 2031 of *Lecture Notes in Computer Science*, pages 268–283. Springer, 2001.

[5] Armin Biere, Alessandro Cimatti, Edmund M. Clarke Jr., Ofer Strichman, and Yunshan Zhu. Bounded model checking. *Advances in Computers*, 58:117–148, 2003.

[6] Randal E. Bryant. Graph-based algorithms for boolean function manipulation. *IEEE Trans. Computers*, 35(8):677–691, 1986.

[7] Janusz A. Brzozowski. Derivatives of regular expressions. *J. ACM*, 11(4):481–494, 1964.

[8] Roy H. Campbell and A. Nico Habermann. The specification of process synchronization by path expressions. In *Symposium on Operating Systems*, volume 16 of *Lecture Notes in Computer Science*, pages 89–102. Springer, 1974.

[9] Edmund M. Clarke Jr. and E. Allen Emerson. Design and synthesis of synchronization skeletons using branching-time temporal logic. In *Logic of Programs*, volume 131 of *Lecture Notes in Computer Science*, pages 52–71. Springer, 1981.

[10] Edmund M. Clarke Jr., Thomas A. Henzinger, Helmut Veith, and Roderick Bloem, editors. *Handbook of Model Checking*. Springer, 2018.

[11] Edmund M. Clarke Jr., William Klieber, Milos Novácek, and Paolo Zuliani. Model checking and the state explosion problem. In *LASER Summer School*, volume 7682 of *Lecture Notes in Computer Science*, pages 1–30. Springer, 2011.

[12] Patrick Cousot. *Méthodes itératives de construction et d'approximation de points fixes d'opérateurs monotones sur un treillis, analyse sémantique de programmes (in French)*. Thèse d'État ès sciences mathématiques, Université de Grenoble Alpes, Grenoble, France, 21 March 1978.

[13] Patrick Cousot. Partial completeness of abstract fixpoint checking. In *SARA*, volume 1864 of *Lecture Notes in Computer Science*, pages 1–25. Springer, 2000.

[14] Patrick Cousot and Radhia Cousot. Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *POPL*, pages 238–252. ACM, 1977.

[15] Patrick Cousot and Radhia Cousot. Constructive versions of Tarski's fixed point theorems. *Pacific Journal of Mathematics*, 81(1):43–57, 1979.

[16] Patrick Cousot and Radhia Cousot. Systematic design of program analysis frameworks. In *POPL*, pages 269–282. ACM Press, 1979.

[17] Patrick Cousot and Radhia Cousot. Temporal abstract interpretation. In *POPL*, pages 12–25. ACM, 2000.

[18] Silvia Crafa, Francesco Ranzato, and Francesco Tapparo. Saving space in a time efficient simulation algorithm. *Fundam. Inform.*, 108(1-2):23–42, 2011.

[19] John Cyphert, Jason Breck, Zachary Kincaid, and Thomas W. Reps. Refinement of path expressions for static analysis. *PACMPL*, 3(POPL):45:1–45:29, 2019.

[20] Giuseppe De Giacomo and Moshe Y. Vardi. Linear temporal logic and linear dynamic logic on finite traces. In *IJCAI*, pages 854–860. IJCAI/AAAI, 2013.

[21] E. Allen Emerson and Joseph Y. Halpern. "sometimes" and "not never" revisited: on branching versus linear time temporal logic. *J. ACM*, 33(1):151–178, 1986.

[22] Mai Gehrke, Daniela Petrisan, and Luca Reggio. Quantifiers on languages and codensity monads. In *LICS*, pages 1–12. IEEE Computer Society, 2017.

[23] Roberto Giacobazzi and Elisa Quintarelli. Incompleteness, counterexamples, and refinements in abstract model-checking. In *SAS*, volume 2126 of *Lecture Notes in Computer Science*, pages 356–373. Springer, 2001.

[24] Roberto Giacobazzi and Francesco Ranzato. Incompleteness of states w.r.t. traces in model checking. *Inf. Comput.*, 204(3):376–407, 2006.

[25] David Harel, Dexter Kozen, and Rohit Parikh. Process logic: Expressiveness, decidability, completeness. *J. Comput. Syst. Sci.*, 25(2):144–170, 1982.

[26] Matthias Heizmann, Jochen Hoenicke, and Andreas Podelski. Software model checking for people who love automata. In *CAV*, volume 8044 of *Lecture Notes in Computer Science*, pages 36–52. Springer, 2013.

[27] Peter Henderson. Finite state modelling in program development. In *Proceedings of the International Conference on Reliable Software*, page 221–227. ACM, April 1975.

[28] Peter Henderson and Peter Quarendon. Finite state testing of structured programs. In *Symposium on Programming*, volume 19 of *Lecture Notes in Computer Science*, pages 72–80. Springer, 1974.

[29] David Hilbert and Wilhelm Ackermann. *Grundzüge der Theoretischen Logik*. Springer, 6 edition, 1928, 1949, reprinted 1959. Engl. trans. "Principles of mathematical logic", Lewis M. Hammond, George G. Leckie, F. Steinhardt, AMS Chelsea Pub., 1958, reprinted 2008.

[30] Martin Hofmann and Wei Chen. Abstract interpretation from Büchi automata. In *CSL-LICS*, pages 51:1–51:10. ACM, 2014.

[31] John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman. *Introduction to automata theory, languages, and computation*. Addison-Wesley, 2 edition, 2003.

[32] James C. King. Symbolic execution and program testing. *Commun. ACM*, 19(7):385–394, 1976.

[33] Stephen Cole Kleene. Representation of events in nerve nets and finite automata. In *Automata Studies*, pages 3—42. Princeton University Press, 1951.

[34] Dénes König. Theory of Finite and Infinite Graphs Birkhäuser, 1990.

[35] Saul A. Kripke.. Semantical considerations on modal logic. *Proceedings of a Colloquium on Modal and Many-Valued Logics, Helsinki, 23–26 August, 1962, Acta Philosophica Fennica*, 16:83–94, 1963.

[36] Orna Kupferman. Automata theory and model checking. In Clarke Jr. et al. [10], pages 107–151.

[37] Orna Lichtenstein and Amir Pnueli. Checking that finite state concurrent programs satisfy their linear specification. In *POPL*, pages 97–107. ACM Press, 1985.

[38] Yannis Mallios, Lujo Bauer, Dilsun Kirli Kaynar, and Jay Ligatti. Enforcing more with less: Formalizing target-aware run-time monitors. In *STM*, volume 7783 of *Lecture Notes in Computer Science*, pages 17–32. Springer, 2012.

[39] Laurent Mauborgne. Binary decision graphs. In *SAS*, volume 1694 of *Lecture Notes in Computer Science*, pages 101–116. Springer, 1999.

[40] Steven P. Miller, Michael W. Whalen, and Darren D. Cofer. Software model checking takes off. *Commun. ACM*, 53(2):58–64, 2010.

[41] Scott Owens, John H. Reppy, and Aaron Turon. Regular-expression derivatives re-examined. *J. Funct. Program.*, 19(2):173–190, 2009.

[42] Amir Pnueli. The temporal logic of programs. In *FOCS*, pages 46–57. IEEE Computer Society, 1977.

[43] Jean-Pierre Queille and Joseph Sifakis. Specification and verification of concurrent systems in CESAR. In *Symposium on Programming*, volume 137 of *Lecture Notes in Computer Science*, pages 337–351. Springer, 1982.

[44] Henry Gordon Rice. Classes of recursively enumerable sets and their decision problems. *Trans. Amer. Math. Soc.*, 74(1):358–366, 1953.

[45] Jacques Sakarovitch. *Elements of Automata Theory.* Cambridge University Press, 2009.

[46] Fred B. Schneider. Enforceable security policies. *ACM Trans. Inf. Syst. Secur.*, 3(1):30–50, 2000.

[47] Meera Sridhar and Kevin W. Hamlen. Model-checking in-lined reference monitors. In *VMCAI*, volume 5944 of *Lecture Notes in Computer Science*, pages 312–327. Springer, 2010.

[48] Alfred Tarski. A lattice theoretical fixpoint theorem and its applications. *Pacific J. of Math.*, 5:285–310, 1955.

[49] Pierre Wolper. Temporal logic can be more expressive. *Information and Control*, 56(1/2):72–99, 1983.