Temporal Abstract Interpretation

Patrick COUSOT and Radhia COUSOT

Département d'informatique
École normale supérieure
45 rue d'Ulm
75230 Paris cedex 05, France
Patrick.Cousot@ens.fr

Laboratoire d'informatique
École polytechnique
91128 Palaiseau cedex, France
rcousot@lix.polytechnique.fr

Abstract

We study the abstract interpretation of temporal calculi and logics in a general syntax, semantics and abstraction independent setting. This is applied to the $\widehat{\mu}^{\star}$-calculus, a generalization of the μ-calculus with new reversal and abstraction modalities as well as a new time-symmetric trace-based semantics. The more classical set-based semantics is shown to be an abstract interpretation of the trace-based semantics which leads to the understanding of model-checking and its application to data-flow analysis as incomplete temporal abstract interpretations. Soundness and incompleteness of the abstractions are discussed. The sources of incompleteness, even for finite systems, are pointed out, which leads to the identification of relatively complete sublogics, à la CTL.

1. Introduction

We apply abstract interpretation to temporal calculi and logics. We consider new calculi/logics which, on one hand, are time-symmetric thanks to a "reversal" temporal operator designed to provide a fully symmetric, hence simpler, treatment of past and future and, on the other hand, mix linear and branching time thanks to state closure modalities. We interpret temporal formulae as sets of infinitary time-symmetric traces discussed in Section 3. This is used in Section 4 to define the trace-based semantics of programs generated by a transition system and in Section 5 to provide the trace-based semantics of the $\widehat{\mu}^{\star}$-calculus and its various subcalculi/logics. Rudiments of abstract interpretation are recalled in Section 6 and used in Section 7 to characterize useful properties of trace-based models such as origin, forward, backward and state closeness. The compositional, generic, syntax, semantics and abstraction independent abstract interpretation of general fixpoint definitions (hence of temporal calculi/logics) is developed in Section 8. Sufficient soundness and completeness hypotheses are stated and corresponding correctness theorems proved. This is used in Section 9 to show that the conventional set-based semantics of temporal calculi/logics with respect to a given transition system is an abstract interpretation of the trace-based semantics, which, in the case of the propositional μ-calculus is complete. However this completeness result does not hold in general, in particular when considering the universal and existential abstractions introduced in Section 10. These abstractions are used in Section 11 to show that model checking is an abstract interpretation of the trace-based semantics of temporal calculi/logics (which, in the case of abstract model checking, is composed with state-based abstractions, as shown in Section 14). These checking abstractions are in general incomplete, even for finite systems, so that the sources of incompleteness are pointed out in Section 12. Complete sub-calculi/logics generalizing CTL are identified in Section 13 for which the respective reasonings on sets of traces and sets of states are equivalent, whereas in general we have a sound approximation only. Application to dataflow analysis and the calculation design of the boolean flow equations by abstract interpretation of the temporal specification is discussed in Section 15, in particular to correct an erroneous claim that classical live variable analysis is unsound. Finally a few research perspectives following from the understanding of model checking as abstract interpretation are discussed the conclusive Section 16.

2. Notations

Set theory. $\{x \in S \mid P(x)\}$ is the subset of elements of S satisfying condition P. S is omitted when clear from P. The characteristic function P maps S into the set $\mathbb{B} \triangleq \{\mathrm{tt}, \mathrm{ff}\}$ of true "tt" and false "ff" booleans. $\triangleq$ is the symbol for "is defined as". The conditional is $(\mathrm{tt}\ ?\ x\ ¿\ y) \triangleq x$ and $(\mathrm{ff}\ ?\ x\ ¿\ y) \triangleq y$. $A \times B \triangleq \{\langle x, y\rangle \mid x \in A \wedge y \in B\}$ is the cartesian product of A by B. $A \mapsto B$ is the set of maps of A into B. $\wp(A)$ is the powerset of A, that is the set of all subsets of the set A, including the empty set $\emptyset$. For a relation $t \in \wp(A \times B)$, we indifferently write $\langle s, s'\rangle \in t$, $t(s, s')$ or $s \xrightarrow{t} s'$. The inverse of t is $t^{-1} \triangleq \{\langle s, s'\rangle \mid \langle s', s\rangle \in t\}$. Using Church's lambda notation $\lambda x \cdot e$, we write f or $f[\bullet]$ for $\lambda x \cdot f(x)$ and $f \circ g \triangleq \lambda x \cdot f(g(x))$. The identity map is $\mathbf{1} \triangleq \lambda x \cdot x$. We let $\mathbb{N}$ and $\mathbb{Z}$ be respectively the set of naturals and integers. $\mathbb{O}$ is the class of ordinals. For a sequence $\sigma \in \mathbb{N} \mapsto S$, $\sigma \in \mathbb{Z} \mapsto S$ or a transfinite one $\sigma \in \mathbb{O} \mapsto S$, we write σ_i for $\sigma(i)$.

Order theory. A *poset* $\langle L, \sqsubseteq\rangle$ is a set equipped with a reflexive, antisymmetric and transitive relation $\sqsubseteq$. The *dual* of a statement on this poset is the statement obtained by replacing $\sqsubseteq$ by its dual (or inverse) $\sqsupseteq$. A poset is a *lattice* $\langle L, \sqsubseteq, \sqcup, \sqcap\rangle$ if and only if (iff) every finite subset $S \subseteq L$ has a least upper bound (lub) $\sqcup S$ and a greatest lower bound (glb) $\sqcap S$. A poset is a *complete lattice* $\langle L, \sqsubseteq, \bot, \top, \bigsqcup, \bigsqcap\rangle$ iff every subset $S \subseteq L$ has a least upper bound (lub) $\bigsqcup S$ so that $\bot = \bigsqcup\emptyset$ is the infimum, $\top = \bigsqcup L$ is the supremum and the greatest lower bound $\bigsqcap S = \bigsqcup\{x \in L \mid$

$\exists y \in S : x \sqsubseteq y\}$ is well-defined. An increasing chain of L is $x \in \mathbb{O} \mapsto L$ such that $\forall \beta, \eta \in \mathbb{O} : (\beta < \eta) \Rightarrow (x_\beta \sqsubseteq x_\eta)$. A *complete partial order* (cpo) is a poset $\langle L, \sqsubseteq, \bot, \sqcup\rangle$ with infimum $\bot$ such that any chain C of L has a lub $\sqcup C$. A *complete boolean lattice* $\langle L, \sqsubseteq, \bot, \top, \sqcup, \sqcap, \neg\rangle$ is a complemented complete lattice. The *complement* $\neg x$ of $x \in L$ such that $x \sqcap \neg x = \bot$ and $x \sqcup \neg x = \top$ satisfies $\neg\neg x = x$. A *complete boolean algebra* is a completely distributive complete boolean lattice. An example of complete boolean algebra is the powerset $\langle \wp(S), \subseteq, \emptyset, S, \cup, \cap, \neg\rangle$ of a set S. If S is a set and $\langle L, \sqsubseteq\rangle$ is a poset (resp. cpo, complete lattice, complete boolean algebra) $\langle S \mapsto L, \dot{\sqsubseteq}\rangle$ is a poset (resp. cpo, complete lattice, complete boolean algebra) for the pointwise ordering $f \dot{\sqsubseteq} g \Leftrightarrow \forall x \in S : f(x) \sqsubseteq g(x)$.

Maps. A map f is *strict* (written $f \in \langle L, \sqsubseteq\rangle \overset{\bot}{\mapsto} \langle M, \preceq\rangle$) if it preserves the infimum: if $\bot$ is the $\sqsubseteq$-infimum of L (it is unique whenever it exists) then $f(\bot)$ is the $\preceq$-infimum of M. The dual notion of *co-strictness* is written $f \in \langle L, \sqsubseteq\rangle \overset{\top}{\mapsto} \langle M, \preceq\rangle$. A map f is *monotone* (written $f \in \langle L, \sqsubseteq\rangle \overset{\text{mon}}{\longmapsto} \langle M, \preceq\rangle$) iff $\forall x, y \in L : x \sqsubseteq y \Rightarrow f(x) \preceq f(y)$ and antitone iff $\forall x, y \in L : x \sqsubseteq y \Rightarrow f(x) \succeq f(y)$. Monotony is self-dual. A map is *continuous* (written $f \in \langle L, \sqsubseteq\rangle \overset{\text{con}}{\longmapsto} \langle M, \preceq\rangle$) iff it preserves existing lubs of increasing chains (i.e. if the increasing chain $x \in \mathbb{O} \mapsto L$ has a lub $\bigsqcup_{\beta\in\mathbb{O}} x_\beta$ then $\{f(x_\beta) \mid \beta \in \mathbb{O}\}$ does have a lub $\bigvee_{\beta\in\mathbb{O}} f(x_i)$ such that $f(\bigsqcup_{\beta\in\mathbb{O}} x_\beta) = \bigvee_{\beta\in\mathbb{O}} f(x_\beta)$). The dual notion is that of *co-continuity* (written $f \in \langle L, \sqsubseteq\rangle \overset{\text{co-con}}{\longmapsto} \langle M, \preceq\rangle$). A map is a *complete join morphism* (written $f \in \langle L, \sqsubseteq\rangle \overset{\sqcup}{\mapsto} \langle M, \preceq\rangle$) iff it preserves existing lubs of any set $X \subseteq L$: if $\bigsqcup X$ exists then $f(\bigsqcup X) = \bigvee f(X)$ is the lub of $f(X) \triangleq \{f(x) \mid x \in X\}$. The dual notion of *complete meet morphism* is written $f \in \langle L, \sqsubseteq\rangle \overset{\sqcap}{\mapsto} \langle M, \preceq\rangle$. A complete join morphism is continuous which implies monotony. If the infimum exists (in which case $\bot = \bigsqcup\emptyset$) then a complete join morphism is strict. A *topological upper closure operator* f on a lattice L is a *join morphism* ($\forall x, y \in L : f(x \sqcup y) = f(x) \sqcup f(y)$, whence monotone), *extensive* ($\forall x \in L : x \sqsubseteq f(x)$) and *idempotent* ($\forall x \in L : f(f(x)) = f(x)$).

Fixpoints. Monotone operators $f \in \langle L, \sqsubseteq\rangle \overset{\text{mon}}{\longmapsto} \langle L, \preceq\rangle$ on cpos $\langle L, \sqsubseteq, \bot, \sqcup\rangle$ (resp. complete lattices $\langle L, \sqsubseteq, \bot, \top, \sqcup, \sqcap\rangle$) have a least fixpoint $\text{lfp}^{\sqsubseteq} f$ (resp. and a greatest fixpoint $\text{gfp}^{\sqsubseteq} f$). By order-theoretic duality, $\text{lfp}^{\sqsubseteq} f = \text{gfp}^{\sqsupseteq} f$ and $\text{gfp}^{\sqsubseteq} f = \text{lfp}^{\sqsupseteq} f$. If $f \in \langle L, \sqsubseteq\rangle \overset{\text{con}}{\longmapsto} \langle L, \sqsubseteq\rangle$ then $\text{lfp}^{\sqsubseteq} f = \bigsqcup_{n\in\mathbb{N}} f^n(\bot)$ where $f^{n+1} = f \circ f^n$ and $f^0 = \mathbf{1}$.

3. Time-symmetric trace-based temporal models

The semantics of a language $\mathfrak{L}$ (either a programming language or a specification language such as a temporal calculus/logic) assigns a temporal model $[\![\pi]\!]$ to each program/formula π of $\mathfrak{L}$. $[\![\bullet]\!] \in \mathfrak{L} \mapsto \mathbb{M}$ where $\mathbb{M}$ is the semantic domain. We describe below the temporal model semantic domain $\mathbb{M}$.

3.1 Temporal models

Informally, *temporal models* are sets of traces that is computation paths considered at a certain time. We need sets because of possible indeterminism. The *discrete time* is chosen in $\mathbb{Z}$. *Traces* $\langle i, \sigma\rangle$ record the present time $i \in \mathbb{Z}$ as well as a computation path σ. This *computation path* $\sigma \in \mathbb{Z} \mapsto \mathbb{S}$ records the past states σ_j at all past times $j < i$, the present state σ_i and the future states σ_j at all future times $j > i$ of the computation. So a path is infinite both in the past and in the future (as opposed to the conventional asymmetric finite past and infinite future [20]). Traditionally, a terminating execution has a final state which is repeated forever. Similarly, a starting execution has an initial state which is repeated forever in the past. The set of *states* is assumed to be given. Formally, we let:

$\mathbb{S}$:	states	$\mathbb{P} \triangleq \mathbb{Z} \mapsto \mathbb{S}$	paths
$\mathbb{T} \triangleq \mathbb{Z} \times \mathbb{P}$	traces	$\mathbb{M} \triangleq \wp(\mathbb{T})$	temporal models

3.2 Basic temporal models

Given a set $S \in \wp(\mathbb{S})$ of states, the *S-state model* $\sigma\{\!|S|\!\}$ is the set of traces the present state of which is in S:

$$\sigma\{\!|S|\!\} \triangleq \{\langle i, \sigma\rangle \in \mathbb{T} \mid \sigma_i \in S\} \qquad S\text{-state model} \qquad \textbf{(1)}$$

Given a transition relation $t \in \wp(\mathbb{S} \times \mathbb{S})$, the *$t$-transition model* $\pi\{\!|t|\!\}$ is the set of traces for which the next step is a transition t: $\pi\{\!|t|\!\} \triangleq \{\langle i, \sigma\rangle \in \mathbb{T} \mid \sigma_i \xrightarrow{t} \sigma_{i+1}\}$.In general, t is chosen to be *total* that is $\forall s \in \mathbb{S} : \exists s' \in \mathbb{S} : t(s, s') \wedge \forall s' \in \mathbb{S} : \exists s \in \mathbb{S} : t(s, s')$. Otherwise we define the *initial state transitions* ${\llcorner t} \triangleq \{\langle s', s'\rangle \mid \forall s \in \mathbb{S} : \neg t(s, s')\}$, the *final state transitions* ${t\lrcorner} \triangleq \{\langle s, s\rangle \mid \forall s' \in \mathbb{S} : \neg t(s, s')\}$ and consider the total transition relation ${\llcorner t} \cup t \cup {t\lrcorner}$ instead of the partial t.

3.3 Basic temporal model transformers

Temporal model transformers $T \in \mathbb{M} \mapsto \mathbb{M}$ map models to models. Monotone transformers $T \in \mathbb{M} \overset{\text{mon}}{\longmapsto} \mathbb{M}$ are used in fixpoint definitions.

Predecessor transformer. The *predecessor* $\oplus\{\!|M|\!\}$ of a model M is M considered at the previous time:

$$\begin{aligned}\oplus\{\!|M|\!\} &\triangleq \{\langle i-1, \sigma\rangle \in \mathbb{T} \mid \langle i, \sigma\rangle \in M\} \qquad \text{predecessor of } M\\ &= \{\langle i, \sigma\rangle \in \mathbb{T} \mid \langle i+1, \sigma\rangle \in M\} \qquad \textbf{(2)}\end{aligned}$$

$\oplus\{\!|\bullet|\!\}$ is $\subseteq$-monotone. The intuition is that a trace $\langle i, \sigma\rangle$ of the predecessor $\oplus\{\!|M|\!\}$ of model M will, at next time, be a trace $\langle i+1, \sigma\rangle$ of M.

Reversal transformer. The *path reversal* $\sigma^{\curvearrowleft} \triangleq \lambda j \cdot \sigma_{-j}$ and the *trace reversal* $\langle i, \sigma\rangle^{\curvearrowleft} \triangleq \langle -i, \sigma^{\curvearrowleft}\rangle$ exchange past and future with respect to the time origin (not with respect to the present time). The *model reversal* is:

$$^{\curvearrowleft}\{\!|M|\!\} \triangleq \{\langle i, \sigma\rangle^{\curvearrowleft} \mid \langle i, \sigma\rangle \in M\} \qquad \text{model reversal of } M \qquad \textbf{(3)}$$

$^{\curvearrowleft}\{\!|\bullet|\!\}$ is $\subseteq$-monotone. Reversal is useful to formalize time-symmetric arguments (e.g. as in [9] that backward program analysis is the reversal of forward program analysis) by considering only the unique reversal operator whereas the argument would have to be repeated for all backward modalities with time-asymmetric temporal models.

State closure transformers. The *universal* (respectively *existential*) *temporal model state closure* $\forall\{\!|M_1, M_2|\!\}$ (resp. $\exists\{\!|M_1, M_2|\!\}$) is the set of traces $\langle i, \sigma\rangle$ of M_1 for which all (resp. some) trace in M_1 with the same present state σ_i belongs to M_2. This generalizes the $\forall$ and $\exists$ temporal operators of CTL* [2, 3, 14]. We formally define the *state projection* $\bullet_{\downarrow\bullet} \in \mathbb{M} \times \mathbb{S} \mapsto \mathbb{M}$ as:

$$M_{\downarrow s} \triangleq \{\langle i, \sigma\rangle \in M \mid \sigma_i = s\} \qquad \text{state projection} \qquad \textbf{(4)}$$

so that the *state closure transformers* are:

$$\forall\{\!|M_1, M_2|\!\} \triangleq \{\langle i, \sigma\rangle \in M_1 \mid M_{1\downarrow\sigma_i} \subseteq M_2\} \qquad \text{universal} \quad \textbf{(5)}$$

$$\exists\{\!|M_1, M_2|\!\} \triangleq \{\langle i, \sigma\rangle \in M_1 \mid (M_{1\downarrow\sigma_i} \cap M_2) \neq \emptyset\} \qquad \text{existential}$$

These transformers are dual in that $\forall\{\!|M_1, M_2|\!\} = \neg\exists\{\!|M_1, \neg M_2|\!\}$ and $\exists\{\!|M_1, M_2|\!\} = \neg\forall\{\!|M_1, \neg M_2|\!\}$. $\forall\{\!|M_1, M_2|\!\}$ and $\exists\{\!|M_1, M_2|\!\}$ are both $\subseteq$-monotone in M_2.

Union, intersection and complement transformers. The *union* and *intersection* of models are monotone while the *complement* is antitone (i.e. $M_1 \subseteq M_2 \Rightarrow \neg\{\!|M_1|\!\} \supseteq \neg\{\!|M_2|\!\}$):

$$\cup\{\!|M_1, M_2|\!\} \triangleq \{\langle i, \sigma\rangle \mid (\langle i, \sigma\rangle \in M_1) \vee (\langle i, \sigma\rangle \in M_2)\} \qquad \text{union}$$

$$\neg\{\!|M|\!\} \triangleq \{\langle i, \sigma\rangle \mid \langle i, \sigma\rangle \notin M\} \qquad \text{complement} \quad \textbf{(6)}$$

$$\cap\{\!|M_1, M_2|\!\} \triangleq \neg\{\!|\cup\{\!|\neg\{\!|M_1|\!\}, \neg\{\!|M_2|\!\}|\!\}|\!\} \qquad \text{intersection}$$

3.4 Temporal model fixpoint definition

Temporal models can be defined as the least fixpoint $\text{lfp}^{\subseteq} T$ (resp. greatest fixpoint $\text{gfp}^{\subseteq} T$) of $\subseteq$-monotone temporal model transformers $T \in \mathbb{M} \overset{\text{mon}}{\longmapsto} \mathbb{M}$.

3.5 Derived temporal model transformers

The classical past and future modal operators of [20] can all be defined in terms of the basic models and transformer fixpoints (including the new state abstraction and reversal transformers), but with a different trace-based semantics. For example, the *successor* $\ominus\{\!|M|\!\}$ of a model M is M considered at the next time:

$$\begin{aligned}\ominus\{\!|M|\!\} &\triangleq {\curvearrowleft}\{\!|\oplus\{\!|{\curvearrowleft}\{\!|M|\!\}|\!\}|\!\} = \{\langle i+1, \sigma\rangle \in \mathbb{T} \mid \langle i, \sigma\rangle \in M\}\\ &= \{\langle i, \sigma\rangle \in \mathbb{T} \mid \langle i-1, \sigma\rangle \in M\} \qquad \text{successor of } M \quad \textbf{(7)}\end{aligned}$$

(this definition would not hold for time-asymmetric temporal models [20]). $\uplus\{\!|M_1, M_2|\!\}$ is the model such that, while future time passes away, all traces are in M_1 until they are in M_2:

$$\begin{aligned}\uplus\{\!|M_1, M_2|\!\} &\triangleq \text{lfp}^{\subseteq} \lambda X \cdot \cup\{\!|M_2, \cap\{\!|M_1, \oplus\{\!|X|\!\}|\!\}|\!\} \qquad \text{until}\\ &= \{\langle i, \sigma\rangle \in \mathbb{T} \mid \exists k \geq i : \langle k, \sigma\rangle \in M_2 \wedge\\ &\qquad\qquad \forall j : i \leq j < k : \langle j, \sigma\rangle \in M_1\}\end{aligned}$$

Its time-symmetric version "since" is $\reflectbox{$\uplus$}\{\!|M_1, M_2|\!\} \triangleq {\curvearrowleft}\{\!|\uplus\{\!|{\curvearrowleft}\{\!|M_1|\!\}, {\curvearrowleft}\{\!|M_2|\!\}|\!\}|\!\}$. We define the following model transformers:

$$\begin{aligned}\diamondplus\{\!|M|\!\} &\triangleq \uplus\{\!|\mathbb{T}, M|\!\} && \text{sometime}\\ \boxplus\{\!|M|\!\} &\triangleq \neg\{\!|\diamondplus\{\!|\neg\{\!|M|\!\}|\!\}|\!\} && \text{always}\\ \boxminus\{\!|M|\!\} &\triangleq {\curvearrowleft}\{\!|\boxplus\{\!|{\curvearrowleft}\{\!|M|\!\}|\!\}|\!\} && \text{has always been}\\ &= \text{gfp}^{\subseteq} \lambda X \cdot \cap\{\!|M, \ominus\{\!|X|\!\}|\!\} && \textbf{(8)}\\ \boxed{\pm}\{\!|M|\!\} &\triangleq \cap\{\!|\boxplus\{\!|M|\!\}, \boxminus\{\!|M|\!\}|\!\} && \text{ever} \quad \textbf{(9)}\end{aligned}$$

4. Program semantics

Following [9], we assume that the program semantics is given by a transition system $\langle \mathbb{S}, \tau\rangle$ where $\mathbb{S}$ is a set of states and the transition relation $\tau \in \wp(\mathbb{S} \times \mathbb{S})$ is described e.g. by a small-step operational semantics.

Hypothesis (10) [total transition system] *$\langle \mathbb{S}, \tau\rangle$ is a total transition system.*

Definition (11) [model generated by the transition system] *The temporal model $\mathcal{M}_\tau$ generated by the program/transition system $\langle \mathbb{S}, \tau\rangle$ is:*

$$\mathcal{M}_\tau \triangleq \boxplus\{\!|\pi\{\!|\tau|\!\}|\!\} = \{\langle i, \sigma\rangle \in \mathbb{T} \mid \forall k \in \mathbb{Z} : \sigma_k \xrightarrow{\tau} \sigma_{k+1}\}$$

Lemma (12) [totality of the model] *$\mathcal{M}_\tau$ is* total *i.e.* $\forall s \in \mathbb{S} : \mathcal{M}_{\tau\downarrow s} \neq \emptyset$.

5. The syntax and semantics of modal calculi and logics

We consider various generalizations of classical modal calculi and logics, with a different trace-based semantics and including new modalities such as state abstraction and reversal.

The reversible $\overset{\curvearrowleft}{\mu}$-calculus is inspired from Kozen's [18] μ-calculus. Assume that $\mathbb{X}$ is an infinite set of *variables*. An *environment* $\rho \in \mathbb{E} \triangleq \mathbb{X} \mapsto \mathbb{M}$ assigns a model $\rho(X)$ to free variables $X \in \mathbb{X}$ of a temporal formula. The *substitution* $\rho[X := x]$ is ρ' such that $\rho'(X) = x$ and $\rho'(Y) = \rho(Y)$ when $Y \neq X$. The *semantics* $[\![\varphi]\!] \in \mathbb{E} \mapsto \mathbb{M}$ of formula φ of the $\overset{\curvearrowleft}{\mu}$-calculus maps environments to the model described by that formula φ. The semantics is partially defined by structural induction on φ:

Definition (13) [semantics of the $\overset{\curvearrowleft}{\mu}^\star$-calculus]

$$\begin{array}{lll}\varphi ::= \sigma_S & S \in \wp(\mathbb{S}) & [\![\sigma_S]\!]\rho \triangleq \sigma\{\!|S|\!\}\\ \quad \mid \pi_t & t \in \wp(\mathbb{S} \times \mathbb{S}) & [\![\pi_t]\!]\rho \triangleq \pi\{\!|t|\!\}\\ \quad \mid \oplus\varphi_1 & & [\![\oplus\varphi_1]\!]\rho \triangleq \oplus\{\!|[\![\varphi_1]\!]\rho|\!\}\\ \quad \mid \varphi_1{}^{\curvearrowleft} & & [\![\varphi_1{}^{\curvearrowleft}]\!]\rho \triangleq {\curvearrowleft}\{\!|[\![\varphi_1]\!]\rho|\!\}\\ \quad \mid \varphi_1 \vee \varphi_2 & & [\![\varphi_1 \vee \varphi_2]\!]\rho \triangleq \cup\{\!|[\![\varphi_1]\!]\rho, [\![\varphi_2]\!]\rho|\!\}\\ \quad \mid \neg\varphi_1 & & [\![\neg\varphi_1]\!]\rho \triangleq \neg\{\!|[\![\varphi_1]\!]\rho|\!\}\\ \quad \mid X & X \in \mathbb{X} & [\![X]\!]\rho \triangleq \rho(X)\\ \quad \mid \mu X \cdot \varphi_1 & & [\![\mu X \cdot \varphi_1]\!]\rho \triangleq \text{lfp}^{\subseteq} \lambda x \cdot [\![\varphi_1]\!]\rho[X := x]\\ \quad \mid \nu X \cdot \varphi_1 & & [\![\nu X \cdot \varphi_1]\!]\rho \triangleq \text{gfp}^{\subseteq} \lambda x \cdot [\![\varphi_1]\!]\rho[X := x]\\ \quad \mid \forall\varphi_1 : \varphi_2 & & [\![\forall\varphi_1 : \varphi_2]\!]\rho \triangleq \forall\{\!|[\![\varphi_1]\!]\rho, [\![\varphi_2]\!]\rho|\!\}\end{array}$$

σ_S is a shorthand for atomic propositions p found in Kripke structures together with the mapping L taking each proposition to some subset of $\mathbb{S}$ (the states where the proposition is true) in that $S = L(p)$. Missing operators are abbreviations (such as e.g. $\exists\varphi_1 : \varphi_2 \triangleq \neg(\forall\varphi_1 : (\neg\varphi_2))$, $\varphi_1 \wedge \varphi_2 \triangleq \neg(\neg\varphi_1 \vee \neg\varphi_2)$, etc). We write $\varphi_1 \equiv \varphi_2 \triangleq \forall\rho \in \mathbb{E} : [\![\varphi_1]\!]\rho = [\![\varphi_2]\!]\rho$ for *semantic equivalence.*

The reversible $\overset{\curvearrowleft}{\text{CTL}}^\star$ temporal logic is inspired from CTL* [2, 3, 14]. It includes the reversal operator and the semantics $[\![\varphi]\!] \in \mathbb{M}$ of $\varphi \in \overset{\curvearrowleft}{\text{CTL}}^\star$ is that $[\![\varphi]\!]\emptyset$ of the $\overset{\curvearrowleft}{\mu}$-calculus in the empty environment $\emptyset$. (The "until" modal operator is $\varphi_1 \,\mathbf{U}\, \varphi_2 \triangleq \mu X \cdot \varphi_2 \vee (\varphi_1 \wedge \oplus X)$ where $X \notin FV(\varphi_1) \cup FV(\varphi_2)$):

$$\varphi ::= \sigma_S \mid \pi_t \mid \oplus\varphi_1 \mid \varphi_1{}^{\curvearrowleft} \mid \varphi_1 \vee \varphi_2 \mid \neg\varphi_1 \mid \varphi_1 \,\mathbf{U}\, \varphi_2 \mid \forall\varphi_1$$

The universal state closure $\forall\varphi_1$ is with respect to a transition system $\langle \mathbb{S}, \tau\rangle$ which, as in the case of CTL*, is left implicit. So $\forall\varphi_1 \triangleq \forall\boxplus(\pi_\tau) : \varphi_1$. $\text{CTL}^\star_+$ is the forward $\overset{\curvearrowleft}{\text{CTL}}^\star$, that is without the reversal operator. The backward $\mathrm{CTL}^\star_-$ is $\overset{\curvearrowleft}{\mathrm{CTL}}{}^\star$ without the reversal operator and the primitive operators $\boldsymbol{\pi}_t$, $\oplus$ and $\mathbf{U}$ replaced by their respective time-symmetric versions $\boldsymbol{\pi}_{t-1}$, $\ominus$ and $\mathbf{S}$ (such that $\ominus\varphi_1 \stackrel{\Delta}{=} (\oplus\varphi_1{}^{\curvearrowleft})^{\curvearrowleft}$ and $\varphi_1\,\mathbf{S}\,\varphi_2 \stackrel{\Delta}{=} (\varphi_1{}^{\curvearrowleft}\,\mathbf{U}\,\varphi_2{}^{\curvearrowleft})^{\curvearrowleft}$). The forward/backward terminology will be justified in Section 7.

The reversible $\overset{\curvearrowleft}{\mathbf{CTL}}$ temporal logic is inspired from CTL [2, 3, 14]. It semantics is that of $\overset{\curvearrowleft}{\mathrm{CTL}}{}^\star$ but the syntax is restricted as follows (the "unless" modality is $\varphi_1\,\mathbf{V}\,\varphi_2 \stackrel{\Delta}{=} \nu X \cdot \varphi_2 \vee (\varphi_1 \wedge \oplus X)$ where $X \notin FV(\varphi_1) \cup FV(\varphi_2)$):

$$\begin{array}{lll} \psi & ::= \boldsymbol{\sigma}_S \mid \psi_1 \vee \psi_2 \mid \psi_1 \wedge \psi_2 \mid \neg\psi_1 \mid \forall\varphi \mid \exists\varphi & \text{state formulae} \\ \varphi & ::= \psi \mid \boldsymbol{\pi}_t \mid \oplus\psi \mid \psi_1\,\mathbf{U}\,\psi_2 \mid \psi_1\,\mathbf{V}\,\psi_2 \mid \varphi_1{}^{\curvearrowleft} & \text{path formulae} \end{array}$$

$\forall\overset{\curvearrowleft}{\mathrm{CTL}}$ (respectively $\exists\overset{\curvearrowleft}{\mathrm{CTL}}$) is $\overset{\curvearrowleft}{\mathrm{CTL}}$ without $\forall$ (resp. $\exists$) operator. The forward logics CTL_+, $\forall\mathrm{CTL}_+$ and $\exists\mathrm{CTL}_+$ are $\overset{\curvearrowleft}{\mathrm{CTL}}$, $\forall\overset{\curvearrowleft}{\mathrm{CTL}}$ and $\exists\overset{\curvearrowleft}{\mathrm{CTL}}$ without the reversal operator. The corresponding backward logics are respectively CTL_-, $\forall\mathrm{CTL}_-$ and $\exists\mathrm{CTL}_-$.

6. Rudiments of abstract interpretation

We introduce abstract interpretation [8, 9] very briefly. We use the Galois connection-based presentation (but closure operators, Moore families, etc. would be equivalent formalizations [9]). A pair of maps $\alpha \in M \mapsto L$ and $\gamma \in L \mapsto M$ between the posets $\langle M, \preceq\rangle$ and $\langle L, \sqsubseteq\rangle$ is a *Galois connection* iff:

$$\forall x \in M, y \in L : \alpha(x) \sqsubseteq y \Leftrightarrow x \preceq \gamma(y).$$

This is denoted $\langle M, \preceq\rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle L, \sqsubseteq\rangle$. α is called the *abstraction* and γ the *concretization*. This definition is equivalent to the four conditions (a) α is monotone, (b) γ is monotone, (c) $\mathbf{1} \mathrel{\dot{\preceq}} \gamma\circ\alpha$ and (d) $\alpha\circ\gamma \mathrel{\dot{\sqsubseteq}} \mathbf{1}$. We have $\alpha\circ\gamma\circ\alpha = \alpha$ and $\gamma\circ\alpha\circ\gamma = \gamma$. If M has an infimum 0 then $\alpha(0)$ is the infimum of L (since $\forall x \in M : 0 \preceq \gamma(x)$ so $\alpha(0) \sqsubseteq x$) and if L has a supremum $\top$ then $\gamma(T)$ is the supremum of M (again $\forall x \in M : \alpha(x) \sqsubseteq \top$ so $x \preceq \gamma(\top)$). This also follows from the fact that α preserves existing lubs while γ preserves existing glbs. The composition of Galois connections $\langle M, \preceq\rangle \underset{\alpha_1}{\overset{\gamma_1}{\leftrightarrows}} \langle L, \sqsubseteq\rangle$ and $\langle L, \sqsubseteq\rangle \underset{\alpha_2}{\overset{\gamma_2}{\leftrightarrows}} \langle N, \leq\rangle$ is a Galois connection $\langle M, \preceq\rangle \underset{\alpha_2\circ\alpha_1}{\overset{\gamma_1\circ\gamma_2}{\leftrightarrows}} \langle N, \leq\rangle$.

We write $\langle M, \preceq\rangle \underset{\alpha}{\overset{\gamma}{\substack{\longleftarrow\\\twoheadrightarrow}}} \langle L, \sqsubseteq\rangle$ when α is surjective (or equivalently γ is injective or $\alpha\circ\gamma = \mathbf{1}$), $\langle M, \preceq\rangle \underset{\alpha}{\overset{\gamma}{\substack{\twoheadleftarrow\\\longrightarrow}}} \langle L, \sqsubseteq\rangle$ when α is injective (or γ is surjective or $\gamma\circ\alpha = \mathbf{1}$) and $\langle M, \preceq\rangle \underset{\alpha}{\overset{\gamma}{\substack{\twoheadleftarrow\\\twoheadrightarrow}}} \langle L, \sqsubseteq\rangle$ when α is bijective. If α is surjective and M has a supremum 1 then $\alpha(1)$ is the supremum of L (since $\forall y \in L : \exists x \in M : \alpha(x) = y$ so $x \preceq 1$ implies by monotony that $y = \alpha(x) \sqsubseteq \alpha(1)$.

If S is a set and $\langle M, \preceq\rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle L, \sqsubseteq\rangle$ then $\langle S \mapsto M, \dot{\preceq}\rangle \underset{\dot{\alpha}}{\overset{\dot{\gamma}}{\leftrightarrows}} \langle S \mapsto L, \dot{\sqsubseteq}\rangle$ where $\dot{\alpha}(f) \stackrel{\Delta}{=} \alpha\circ f$ and $\dot{\gamma}(g) = \gamma\circ g$. We also have $\langle (S \mapsto M) \overset{\text{mon}}{\longmapsto} M, \ddot{\preceq}\rangle \underset{\vec{\alpha}}{\overset{\vec{\gamma}}{\leftrightarrows}} \langle (S \mapsto L) \overset{\text{mon}}{\longmapsto} L, \ddot{\sqsubseteq}\rangle$ with $f \mathrel{\ddot{\preceq}} g \stackrel{\Delta}{=} \forall x : f(x) \leq g(x)$, $\vec{\alpha}(F) \stackrel{\Delta}{=} \alpha\circ F\circ\dot{\gamma}$ and $\vec{\gamma}(G) = \gamma\circ G\circ\dot{\alpha}$. Moreover if α is surjective then so are $\dot{\alpha}$ and $\vec{\alpha}$.

If $\langle L, \preceq, 0, 1, \vee, \wedge, \neg\rangle$ and $\langle M, \sqsubseteq, \bot, \top, \sqcup, \sqcap, \neg\rangle$ are complete boolean algebras and $f \in L \mapsto M$ then we define the *left complement* $\urcorner f \stackrel{\Delta}{=} \lambda x \cdot \neg f(x)$, the *right complement* $f\urcorner \stackrel{\Delta}{=} \lambda x \cdot f(\neg x)$ and the *dual* $\widetilde{f} \stackrel{\Delta}{=} \urcorner f \urcorner = \lambda M \cdot \neg f(\neg M)$ of f. If $\langle L, \preceq\rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle M, \sqsubseteq\rangle$ then $\langle L, \succeq\rangle \underset{\alpha\urcorner}{\overset{\urcorner\gamma}{\leftrightarrows}} \langle M, \sqsubseteq\rangle$ is the right complement Galois connection, $\langle L, \preceq\rangle \underset{\urcorner\alpha}{\overset{\gamma\urcorner}{\leftrightarrows}} \langle M, \sqsupseteq\rangle$ is the left complement Galois connection, and $\langle L, \succeq\rangle \underset{\widetilde{\alpha}}{\overset{\widetilde{\gamma}}{\leftrightarrows}} \langle M, \sqsubseteq\rangle$ is the dual Galois connection (and similarly for Galois injections/surjections).

Fundamental results of abstract interpretation theory [6, 9] include fixpoint transfer and approximation theorems.

Proposition (14) [Least fixpoint image] *If $\langle M, \preceq, 0, \vee\rangle$ is a cpo, the abstraction $\alpha \in \langle M, \preceq\rangle \overset{\bot,con}{\longmapsto} \langle L, \sqsubseteq\rangle$ is strict and continuous, $\mathcal{F} \in M \overset{mon}{\longmapsto} M$ and $\mathcal{G} \in L \overset{mon}{\longmapsto} L$ are monotone and satisfy the* α-commutation condition $\alpha\circ\mathcal{F} = \mathcal{G}\circ\alpha$ *then the least fixpoints are well-defined such that* $\alpha(\mathrm{lfp}^{\preceq}\mathcal{F}) = \mathrm{lfp}^{\sqsubseteq}\mathcal{G}$.

In particular $\langle M, \preceq\rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle L, \sqsubseteq\rangle$ implies $\alpha \in \langle M, \preceq\rangle \overset{\bot,\mathrm{con}}{\longmapsto} \langle L, \sqsubseteq\rangle$ in which case Proposition (14) is directly applicable. For the greatest fixpoint image, we use the dual of Proposition (14).

The same way, if $\langle L, \sqsubseteq, \top, \sqcap\rangle$ is a dual cpo and $\langle M, \preceq\rangle \underset{\alpha}{\overset{\gamma}{\substack{\longleftarrow\\\twoheadrightarrow}}} \langle L, \sqsubseteq\rangle$ is a Galois surjection then $\langle L, \sqsupseteq\rangle \underset{\gamma}{\overset{\alpha}{\substack{\twoheadleftarrow\\\longrightarrow}}} \langle M, \succeq\rangle$ so that if $\mathcal{F} \in M \overset{\text{mon}}{\longmapsto} M$ and $\mathcal{G} \in L \overset{\text{mon}}{\longmapsto} L$ are monotone and satisfy the *γ-commutation condition* $\mathcal{F}\circ\gamma = \gamma\circ\mathcal{G}$ then, by the dual of Proposition (14), the fixpoints are well-defined such that $\gamma(\mathrm{lfp}^{\sqsupseteq}\mathcal{G}) = \mathrm{lfp}^{\succeq}\mathcal{F}$ so $\alpha(\mathrm{gfp}^{\preceq}\mathcal{F}) = \alpha\circ\gamma(\mathrm{gfp}^{\sqsubseteq}\mathcal{G}) = \mathrm{gfp}^{\sqsubseteq}\mathcal{G}$ since $\alpha\circ\gamma = \mathbf{1}$.

The α- and γ-commutation conditions are independent, including for Galois surjections.

Finally, if $\langle M, \preceq, 0, 1, \vee, \wedge, \neg\rangle$ and $\langle L, \sqsubseteq, \bot, \top, \sqcup, \sqcap, \neg\rangle$ are complete boolean lattices and $\langle M, \preceq\rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle L, \sqsubseteq\rangle$ then $\langle M, \succeq\rangle \underset{\widetilde{\alpha}}{\overset{\widetilde{\gamma}}{\leftrightarrows}} \langle L, \sqsupseteq\rangle$ so that if $\mathcal{F} \in M \overset{\text{mon}}{\longmapsto} M$ and $\mathcal{G} \in L \overset{\text{mon}}{\longmapsto} L$ are monotone and satisfy the *dual commutation condition* $\widetilde{\alpha}\circ\mathcal{F} = \mathcal{G}\circ\widetilde{\alpha}$ then, by the dual of Proposition (14), the greatest fixpoints are well-defined such that $\widetilde{\alpha}(\mathrm{gfp}^{\preceq}\mathcal{F}) = \mathrm{gfp}^{\sqsubseteq}\mathcal{G}$

As a direct application of Proposition (14), consider a monotone operator $f \in L \overset{\text{mon}}{\longmapsto} L$ on a complete boolean lattice L. We have $\langle L, \sqsubseteq\rangle \underset{\neg}{\overset{\neg}{\substack{\twoheadleftarrow\\\twoheadrightarrow}}} \langle L, \sqsupseteq\rangle$ and $\neg\circ f = \widetilde{f}\circ\neg$ so that Proposition (14) implies Park's *complement duality fixpoint theorem*:

$$\neg\mathrm{lfp}^{\sqsupseteq} f = \neg\mathrm{gfp}^{\sqsubseteq} f = \mathrm{lfp}^{\sqsubseteq}\widetilde{f}\,.$$

The *fixpoint reversal theorem* also follows from the fixpoint transfer Proposition (14) in the cpo $\langle \mathbb{M}, \subseteq, \emptyset, \cup\rangle$ for the Galois isomorphism $\langle \mathbb{M}, \subseteq\rangle \underset{{}^{\curvearrowleft}\llbracket\bullet\rrbracket}{\overset{{}^{\curvearrowleft}\llbracket\bullet\rrbracket}{\substack{\twoheadleftarrow\\\twoheadrightarrow}}} \langle \mathbb{M}, \subseteq\rangle$ and states that:

$${}^{\curvearrowleft}\llbracket\mathrm{lfp}^{\subseteq} T\rrbracket = \mathrm{lfp}^{\subseteq}\lambda X \cdot {}^{\curvearrowleft}\llbracket T\, {}^{\curvearrowleft}\llbracket X\rrbracket\rrbracket$$

Proposition (15) [Least fixpoint approximation] *If $\langle M, \preceq, 0, \vee\rangle$ and $\langle L, \sqsubseteq, \bot, \sqcup\rangle$ are cpos, the abstraction $\alpha \in \langle M, \preceq\rangle \overset{\bot,con}{\longmapsto} \langle L, \sqsubseteq\rangle$ is strict and continuous, $\mathcal{F} \in M \overset{mon}{\longmapsto} M$ and $\mathcal{G} \in L \overset{mon}{\longmapsto} L$ are monotone and satisfy the* α-semi-commutation condition $\alpha\circ\mathcal{F} \mathrel{\dot{\sqsubseteq}} \mathcal{G}\circ\alpha$ *then the least fixpoints are well-defined such that:* $\alpha(\mathrm{lfp}^{\preceq}\mathcal{F}) \sqsubseteq \mathrm{lfp}^{\sqsubseteq}\mathcal{G}$.

Proposition (16) [Greatest fixpoint approximation] *If $\langle M, \preceq, 1, \wedge\rangle$ and $\langle L, \sqsubseteq, \top, \sqcap\rangle$ are dual cpos, the abstraction $\alpha \in \langle M, \preceq\rangle \overset{mon}{\longmapsto} \langle L, \sqsubseteq\rangle$ is monotone, $\mathcal{F} \in M \overset{mon}{\longmapsto} M$ and $\mathcal{G} \in L \overset{mon}{\longmapsto} L$ are monotone and satisfy the* α-semi-commutation condition $\alpha\circ\mathcal{F} \mathrel{\dot{\sqsubseteq}} \mathcal{G}\circ\alpha$ *then the greatest fixpoints are well-defined such that* $\alpha(\mathrm{gfp}^{\preceq}\mathcal{F}) \sqsubseteq \mathrm{gfp}^{\sqsubseteq}\mathcal{G}$.

In the particular case when $\langle M, \preceq\rangle \xleftrightarrows[\alpha]{\gamma} \langle L, \sqsubseteq\rangle$ which implies $\alpha \in \langle M, \preceq\rangle \overset{\text{mon}}{\longmapsto} \langle L, \sqsubseteq\rangle$, the semi-commutation condition $\alpha \circ \mathcal{F} \mathrel{\dot{\sqsubseteq}} \mathcal{G} \circ \alpha$ is equivalent to $\mathcal{F} \circ \gamma \mathrel{\dot{\preceq}} \gamma \circ \mathcal{G}$.

7. Origin/forward/backward/state closed temporal formulae

We define origin-closed temporal models (invariant by shifting the origin of time), forward-closed temporal models (referring only to the future), backward-closed models (referring only to the past) and state closed models (referring only to the present). We propose an abstract interpretation of the concrete semantics of the temporal formulae supplying a sound criterion for checking forward/backward closeness. Besides providing a simple instantiation of Section 6, this abstract interpretation will be useful in the forthcoming Section 13 when discussing the completeness of the transition checking abstraction of temporal calculi/logics.

7.1 Origin closeness

A temporal model $M \in \mathbb{M}$ is *origin closed* if the present time can be shifted arbitrarily along any trace of M without changing the model M. We define $Or(M) \triangleq \{\langle i+k, \lambda j \cdot \sigma_{j-k}\rangle \mid \langle i, \sigma\rangle \in M \wedge k \in \mathbb{Z}\}$. $Or(\bullet)$ is a topological operator so that *origin independent models* M satisfy $M \subseteq Or(M)$ or equivalently $Or(M) = M$. In general temporal models are <u>not</u> origin independent. However:

Lemma (17) *The temporal model $\mathcal{M}_\tau$ generated by a transition system $\langle \mathbb{S}, \tau\rangle$ is origin independent.*

7.2 Forward/backward state closeness

If $\eta, \sigma \in \mathbb{P}$ are paths then the *prolongation of η by σ at/after time $i \in \mathbb{Z}$* is:

$$\eta[_i\sigma \triangleq \lambda k \cdot (k < i \;?\; \eta_k \;¿\; \sigma_k) \qquad \text{prolongation at } i$$

$$\eta_i]\sigma \triangleq (\sigma^\frown[_{-i}\eta^\frown)^\frown = \lambda k \cdot (k \leq i \;?\; \eta_k \;¿\; \sigma_k) \qquad \text{prolong. after } i$$

A temporal model $M \in \mathbb{M}$ is *forward closed* if the past of each trace of M can be changed arbitrarily without changing the model M. *Backward closeness* is the time symmetric notion, so the future is unknown. *State closeness* is when the present only matters. We define:

$$\begin{aligned} Fd(M) &\triangleq \{\langle i, \eta[_i\sigma\rangle \mid \langle i, \sigma\rangle \in M \wedge \eta \in \mathbb{P}\} && \text{forward closure}\\ Bd(M) &\triangleq {}^\frown\{\!|Fd({}^\frown\{\!|M|\!\})|\!\} && \text{backward closure}\\ &= \{\langle i, \sigma_i]\eta\rangle \mid \langle i, \sigma\rangle \in M \wedge \eta \in \mathbb{P}\} \\ St(M) &\triangleq Fd(M) \cup Bd(M) && \text{state closure}\\ &= \{\langle i, \eta[_i\sigma_i]\eta'\rangle \mid \langle i, \sigma\rangle \in M \wedge \eta, \eta' \in \mathbb{P}\} \end{aligned}$$

These are topological upper closure operators on $\mathbb{P}$, so e.g. $Fd(M) \subseteq M$ iff $Fd(M) = M$. We define *forward closeness* $FD(M) \triangleq (Fd(M) \subseteq M)$, *backward closeness* $BD(M) \triangleq (Bd(M) \subseteq M)$ and *state closeness* $ST(M) \triangleq (St(M) \subseteq M)$.

7.3 Forward/backward/state closeness checking

We define an abstract interpretation $\llparenthesis\varphi\rrparenthesis^c \in (\mathbb{X} \mapsto \wp(\mathbb{C})) \mapsto \wp(\mathbb{C})$ where $\mathbb{C} \triangleq \{\mathbf{f}, \mathbf{b}\}$ such that a ground formula φ is forward closed when $\mathbf{f} \in \llparenthesis\varphi\rrparenthesis^c\emptyset$, backward closed when $\mathbf{b} \in \llparenthesis\varphi\rrparenthesis^c\emptyset$ (whence state closed when $\llparenthesis\varphi\rrparenthesis^c\emptyset = \mathbb{C}$). The corresponding abstraction is:

$$\begin{aligned} \alpha^c(X) &\triangleq (\forall M \in X : Fd(M) \subseteq M \;?\; \{\mathbf{f}\} \;¿\; \emptyset) \cup \\ &\qquad\qquad \forall M \in X : Bd(M) \subseteq M \;?\; \{\mathbf{b}\} \;¿\; \emptyset) \\ \gamma^c(C) &\triangleq (\mathbf{f} \in C \;?\; \{M \in \mathbb{M} \mid Fd(M) \subseteq M\} \;¿\; \mathbb{M}) \cap \\ &\qquad\qquad (\mathbf{b} \in C \;?\; \{M \in \mathbb{M} \mid Bd(M) \subseteq M\} \;¿\; \mathbb{M}) \end{aligned}$$

so that $\langle \wp(\mathbb{M}), \subseteq\rangle \xleftrightarrows[\alpha^c]{\gamma^c} \langle \wp(\mathbb{C}), \supseteq\rangle$. For sets of environments, we define:

$$\vec{\alpha}^c(R) \triangleq \lambda X \cdot \alpha^c\{\rho(X) \mid \rho \in R\}$$

$$\vec{\gamma}^c(r) \triangleq \{\rho \mid \forall X \in \mathbb{X} : \rho(X) \in \gamma^c(r(X))\}$$

so that $\langle \wp(\mathbb{E}), \subseteq\rangle \xleftrightarrows[\vec{\alpha}^c]{\vec{\gamma}^c} \langle \wp(\mathbb{X} \mapsto \mathbb{C}), \dot{\supseteq}\rangle$. Finally, for collecting semantics $S \in \wp(\mathbb{E}) \mapsto \wp(\mathbb{M})$, we define:

$$\vec{\vec{\alpha}}^c(S) \triangleq \alpha^c \circ S \circ \vec{\gamma}^c \qquad \vec{\vec{\gamma}}^c(S^c) \triangleq \gamma^c \circ S^c \circ \vec{\alpha}^c$$

so that $\langle \wp(\mathbb{E}) \mapsto \wp(\mathbb{M}), \subseteq\rangle \xleftrightarrows[\vec{\vec{\alpha}}^c]{\vec{\vec{\gamma}}^c} \langle \wp(\mathbb{X} \mapsto \mathbb{C}) \mapsto \wp(\mathbb{C}), \dot{\supseteq}\rangle$.

Given a formula φ, its collecting semantics $\{\!|\varphi|\!\}^c \in \wp(\mathbb{E}) \mapsto \wp(\mathbb{M})$ is $\{\!|\varphi|\!\}^c R \triangleq \{[\![\varphi]\!]\rho \mid \rho \in R\}$ and its abstract closeness semantics $\llparenthesis\varphi\rrparenthesis^c \in \wp(\mathbb{X} \mapsto \mathbb{C}) \mapsto \wp(\mathbb{C})$ should satisfy the following *soundness criterion* $\llparenthesis\varphi\rrparenthesis^c \mathrel{\dot{\subseteq}} \vec{\vec{\alpha}}^c(\{\!|\varphi|\!\}^c)$. This soundness criterion ensures that if $\mathbf{f} \in \llparenthesis\varphi\rrparenthesis^c R$ (resp. $\mathbf{b} \in \llparenthesis\varphi\rrparenthesis^c R$) then for all $\rho \in \vec{\alpha}^c(R)$, $[\![\varphi]\!]\rho$ is a forward (resp. backward) closed temporal model (hence state closed if $\llparenthesis\varphi\rrparenthesis^c R = \mathbb{C}$). The calculational design of the abstract semantics leads to:

$$\llparenthesis\boldsymbol{\pi}_t\rrparenthesis^c R \triangleq (t = \mathbb{S} \times \mathbb{S} \;?\; \mathbb{C} \;¿\; \{\mathbf{f}\}) \qquad \llparenthesis\boldsymbol{\sigma}_S\rrparenthesis^c R \triangleq \{\mathbf{f}, \mathbf{b}\}$$

$$\llparenthesis\oplus\varphi_1\rrparenthesis^c R \triangleq \llparenthesis\varphi_1\rrparenthesis^c R \cap \{\mathbf{f}\} \qquad \llparenthesis\varphi_1{}^\frown\rrparenthesis^c R \triangleq (\llparenthesis\varphi_1\rrparenthesis^c R)^\frown$$

$$\text{where } X^\frown \triangleq \{x^\frown \mid x \in X\}, \mathbf{f}^\frown \triangleq \mathbf{b} \text{ and } \mathbf{b}^\frown \triangleq \mathbf{f}$$

$$\llparenthesis\varphi_1 \vee \varphi_2\rrparenthesis^c R \triangleq \llparenthesis\varphi_1\rrparenthesis^c R \cap \llparenthesis\varphi_2\rrparenthesis^c R \qquad \llparenthesis\neg\varphi_1\rrparenthesis^c R \triangleq \llparenthesis\varphi_1\rrparenthesis^c R$$

$$\llparenthesis\forall\varphi_1 : \varphi_2\rrparenthesis^c R \triangleq \llparenthesis\varphi_1\rrparenthesis^c R \qquad \llparenthesis X\rrparenthesis^c R \triangleq R(X)$$

$$\llparenthesis\boldsymbol{\mu} X \cdot \varphi_1\rrparenthesis^c R \triangleq \llparenthesis\boldsymbol{\nu} X \cdot \varphi_1\rrparenthesis^c R \triangleq \mathrm{gfp}^{\subseteq} \lambda x \cdot \llparenthesis\varphi_1\rrparenthesis^c R[X := x]$$

The abstract closeness semantics of derived temporal operators can easily be calculated. For example: $\llparenthesis\varphi_1 \mathbf{U} \varphi_2\rrparenthesis^c R = \llparenthesis\boldsymbol{\mu} X \cdot \varphi_2 \vee (\varphi_1 \wedge \oplus X)\rrparenthesis^c R = \mathrm{gfp}^{\subseteq} \lambda x \cdot \llparenthesis\varphi_2\rrparenthesis^c R \cap \llparenthesis\varphi_1\rrparenthesis^c R \cap x \cap \{\mathbf{f}\} = \llparenthesis\varphi_1\rrparenthesis^c R \cap \llparenthesis\varphi_2\rrparenthesis^c R \cap \{\mathbf{f}\}$. So if φ_1 and φ_2 are both forward closed then so is $\varphi_1 \mathbf{U} \varphi_2$, otherwise the analysis is inconclusive. The same way, $\boxplus\varphi \triangleq \boldsymbol{\sigma}_{\mathbb{S}} \mathbf{U} \varphi$ so $\llparenthesis\boxplus\varphi\rrparenthesis^c R = \llparenthesis\varphi\rrparenthesis^c R \cap \{\mathbf{f}\}$.

The abstract semantics $\llparenthesis\bullet\rrparenthesis^c$ is not complete in that for example if $\mathbb{S} \triangleq \{\circ, \bullet\}$ then neither $\varphi_1 \equiv \ominus\boldsymbol{\sigma}_{\{\circ\}} \wedge \boxplus\boldsymbol{\sigma}_{\{\bullet\}}$ nor $\varphi_2 \equiv \ominus\boldsymbol{\sigma}_{\{\bullet\}} \wedge \boxplus\boldsymbol{\sigma}_{\{\bullet\}}$ is forward (because the state immediately preceding the present state does matter) whereas their disjunction $\varphi_1 \vee \varphi_2 \equiv \boxplus\boldsymbol{\sigma}_{\{\bullet\}}$ is forward but the analysis is inconclusive since $\llparenthesis\varphi_1 \vee \varphi_2\rrparenthesis^c\emptyset = \llparenthesis\varphi_1\rrparenthesis^c\emptyset \cap \llparenthesis\varphi_2\rrparenthesis^c\emptyset = \emptyset$.

Finally, by induction on the syntax of formulae, it is very easy, using the above abstract interpretation, to prove that:

Lemma (18) *All formulae of* $\mathrm{CTL}^\star_+$, *CTL_+, $\forall CTL_+$ and $\exists CTL_+$ (respectively* $\mathrm{CTL}^\star_-$, *CTL_-, $\forall CTL_-$ and $\exists CTL_-$) are forward (resp. backward) closed.*

8. Compositional generic temporal abstract interpretation

In this section we study the sound (and complete) compositional abstract interpretation of temporal calculi/logics in very general terms that is independently of a particular language, semantics and abstraction (which is general enough for this Section 8 not to be limited to temporal calculi/logics).

Compositional abstract interpretation, that is by induction on the syntax of formulae, is in general confined to a particular syntax and semantics. To avoid this difficulty, we consider a generic abstract syntax, covering all particular cases through abbreviations (i.e. syntactic macros). Moreover the concrete and abstract semantics are also generic in that they are parameterized by a semantic domain and semantic transformers to handle basic operators. To be fully general, we must handle various monotony/antitony conditions both on the semantic transformers and the abstractions. For example negation is antitone and can be used to generate monotone/antitone abstractions α, $\neg\alpha$, $\alpha^{\neg}$ and $\widetilde{\alpha}$. So abstract interpretation soundness is expressed using well-definedness and typing conditions in order to cover all cases at once. Moreover soundness and in/completeness are handled together.

8.1 Abstract syntax

We let $\mathcal{O} = \bigcup_{n \geq 0} \mathcal{O}^n$ be the set of operators $\Psi_n \in \mathcal{O}^n$ of arity n. The formulae/sentences φ of the language $\mathcal{L}$ have the following syntax:

Definition (19) [abstract syntax]

$$\begin{array}{llll} \varphi ::= & \Psi_n(\varphi_1, \ldots, \varphi_n) & \Psi_n \in \mathcal{O}^n,\ n \geq 0 & \textit{n-ary operator} \\ \mid & X & X \in \mathbb{X} & \textit{variable} \\ \mid & \mathsf{f}^p X \cdot \varphi_1 & & \textit{fixpoint} \\ p ::= & + \mid - & & \textit{positiveness} \end{array}$$

For constants $\Psi_0 \in \mathcal{O}^0$, we often write Ψ_0 for $\Psi_0()$. In Definition (19), f^+ stands for the least fixpoint μ and f^- for the greatest ν. The free variables $FV(\varphi)$ and bound variables $BV(\varphi)$ of a formula φ are defined inductively as usual.

8.2 Concrete semantic domain

Hypothesis (20) [concrete domain] *The concrete semantic domain* $\langle \mathbb{L}, \leq, 0, 1, \vee, \wedge, \neg \rangle$ *is a complete boolean algebra.*

8.3 Concrete interpretation of primitive operators

In order to reason as if all operators were monotone we introduce the notion of *monotony up to positiveness*. We define $\leq^+ \triangleq \leq$ and $\leq^- \triangleq \geq$ and assign to each operator $\Psi_n \in \mathcal{O}^n$ its *positiveness* $Pos(\!|\Psi_n|\!) \in \{+, -\}^n$. Then the *concrete interpretation* $\mathcal{I}[\![\Psi_n]\!]$ of $\Psi_n \in \mathcal{O}^n$, which belongs to $\mathbb{L}^n \mapsto \mathbb{L}$, satisfies the following monotonicity up to positiveness condition:

Hypothesis (21) [monotonicity of the concrete operators] *If* $Pos(\!|\Psi_n|\!) = \langle p_1, \ldots, p_n \rangle$ *then for all* $x_1, \ldots, x_n, y_1, \ldots, y_n \in \mathbb{L}$:

$$\forall i \in [1, n] : x_i \leq^{p_i} y_i \Rightarrow \mathcal{I}[\![\Psi_n]\!](x_1, \ldots, x_n) \leq \mathcal{I}[\![\Psi_n]\!](y_1, \ldots, y_n)$$

For example $Pos(\!|\neg|\!) = \langle - \rangle$ because $\mathcal{I}[\![\neg]\!] = \neg$ is antitone (by Hypothesis (20), $x \leq^- y \Leftrightarrow x \geq y \Leftrightarrow \neg x \leq \neg y$) while $Pos(\!|\vee|\!) = \langle +, + \rangle$ because $\mathcal{I}[\![\vee]\!] = \sqcup$ is monotone in its two arguments.

8.4 Concrete semantics

Given an environment $\rho \in \mathbb{E}$, the *concrete semantics* of the language $\mathcal{L}$ partially defines the semantic value $[\![\varphi]\!]\rho$ of formula $\varphi \in \mathcal{L}$ in environment ρ. The following compositional definition of the *semantic function* $[\![\bullet]\!] \in \mathcal{L} \mapsto (\mathbb{E} \mapsto \mathbb{L})$ is partial in that the least fixpoint $\mathrm{lfp}^{\leq^+}$ (or the greatest for $\mathrm{lfp}^{\leq^-}$) may not exist:

Definition (22) [concrete semantics]

$$\begin{aligned} [\![\Psi_n(\varphi_1, \ldots, \varphi_n)]\!]\rho &\triangleq \mathcal{I}[\![\Psi_n]\!]([\![\varphi_1]\!]\rho, \ldots, [\![\varphi_n]\!]\rho) \\ [\![X]\!]\rho &\triangleq \rho(X) \\ [\![\mathsf{f}^p X \cdot \varphi_1]\!]\rho &\triangleq \mathrm{lfp}^{\leq^p} \lambda x \cdot [\![\varphi_1]\!]\rho[X := x] \end{aligned}$$

Definition (22) includes the special case $[\![\Psi_0]\!]\rho \triangleq \mathcal{I}[\![\Psi_0]\!]()$ for constants $\Psi_0 \in \mathcal{O}^0$.

8.5 Well-definedness of the concrete semantics

Given an environment $\eta \in \mathbb{X} \mapsto \{+, -\}$, we define the following sufficient *well-definedness condition* [1] $Wd(\!|\varphi|\!)\eta \in \mathbb{B}$ of formulae $\varphi \in \mathcal{L}$ in environment η:

Definition (23) [well-definedness]

$$\begin{aligned} Wd(\!|\Psi_n(\varphi_1, \ldots, \varphi_n)|\!)\eta &\triangleq \bigwedge_{i=1}^{n} Wd(\!|\varphi_i|\!)(\lambda X \cdot p_i \otimes \eta(X)) \\ & \qquad \textbf{where } Pos(\!|\Psi_n|\!) = \langle p_1, \ldots, p_n \rangle \\ Wd(\!|X|\!)\eta &\triangleq (\eta(X) = +) \\ Wd(\!|\mathsf{f}^p X \cdot \varphi_1|\!)\eta &\triangleq Wd(\!|\varphi_1|\!)\eta[X := +] \end{aligned}$$

where the rule of signs is $+ \otimes + \triangleq +$, $+ \otimes - \triangleq -$, $- \otimes + \triangleq -$ *and* $- \otimes - \triangleq +$.

For constants $\Psi_0 \in \mathcal{O}^0$, Definition (23) amounts to $Wd(\!|\Psi_0|\!)\eta \triangleq \mathrm{tt}$.

Example (24) We have $Pos(\!|\neg|\!) = \langle - \rangle$ and $Pos(\!|\vee|\!) = \langle +, + \rangle$ so that the formula $\mu Y \cdot \neg \mu X \cdot \neg(\neg X \vee Y)$ is well-defined, as shown below:

$$\begin{aligned} & Wd(\!|\mu Y \cdot \neg \mu X \cdot \neg(\neg X \vee Y)|\!)\emptyset \\ = \ & Wd(\!|\neg \mu X \cdot \neg(\neg X \vee Y)|\!)\emptyset[Y := +] \\ = \ & Wd(\!|\mu X \cdot \neg(\neg X \vee Y)|\!)\emptyset[Y := -] \\ = \ & Wd(\!|\neg(\neg X \vee Y)|\!)\emptyset[Y := -][X := +] \\ = \ & Wd(\!|\neg X \vee Y|\!)\emptyset[Y := +][X := -] \\ = \ & Wd(\!|\neg X|\!)\emptyset[Y := +][X := -] \wedge Wd(\!|Y|\!)\emptyset[Y := +][X := -] \\ = \ & Wd(\!|X|\!)\emptyset[Y := -][X := +] \wedge Wd(\!|Y|\!)\emptyset[Y := +][X := -] \\ = \ & \emptyset[Y := -][X := +](X) = + \wedge \emptyset[Y := +][X := -](Y) = + \\ = \ & \mathrm{tt} \end{aligned}$$

The following Lemma (25) generalizes the syntactic condition given for the μ-calculus closed formulae stating that variables bound by the fixpoint operators should be in the scope of an even number of negations. It is a special case of Lemma (31) which is proved in Section 8.12.

Lemma (25) *Let* $\varphi \in \mathcal{L}$ *be a formula and* $\eta \in \mathbb{X} \mapsto \{+, -\}$ *such that* $Wd(\!|\varphi|\!)\eta = \mathrm{tt}$. *Let the free variables of* φ *be* $FV(\varphi) = \{X_1, \ldots, X_m\}$. *Then Hypothesis* (21) *implies that for all environments* $\rho \in \mathbb{E}$, *the map* $\lambda x_1, \ldots, x_m \cdot [\![\varphi]\!]\rho[X_1 := x_1, \ldots, X_m := x_m]$ *is well-defined in* $\mathbb{L}^m \mapsto \mathbb{L}$, *monotone in* x_i *if* $\eta(X_i) = +$ *and antitone in* x_i *if* $\eta(X_i) = -$:

$$\begin{aligned} & \forall i \in [1, m] : x_i \leq^{\eta(X_i)} y_i \Rightarrow \\ & \quad [\![\varphi]\!]\rho[X_1 := x_1, \ldots, X_m := x_m] \leq [\![\varphi]\!]\rho[X_1 := y_1, \ldots, X_m := y_m] \end{aligned}$$

In the following we consider formulae $\varphi \in \mathcal{L}$ such that $Wd(\!|\varphi|\!)\eta = \mathrm{tt}$ (also called "in positive form" [18]) which concrete semantics $[\![\varphi]\!]$ is, by Lemma (25), well-defined. Observe that the well-definedness abstract interpretation is sound (in that $Wd(\!|\varphi|\!)\eta = \mathrm{tt}$ implies well-definedness) but incomplete (in that the semantics of φ might be well-defined even when $Wd(\!|\varphi|\!)\eta = \mathrm{ff}$).

[1] We do not develop the point that this well-definedness condition is indeed a typing by abstract interpretation of the semantics with possible undefinedness, see [7].

8.6 Abstract semantic domains

Since in general we use different abstractions (whence abstract domains) for different operators, variables and fixpoints of the language, we let $\mathbb{A}$ be a set of *abstraction indexes* such that the corresponding *abstract domains* $\mathbb{M}^a$, $a \in \mathbb{A}$ satisfy the following:

Hypothesis (26) [abstract domains] *For all $a \in \mathbb{A}$, the a-indexed abstract semantic domain $\langle \mathbb{M}^a, \sqsubseteq^a, \bot^a, \top^a, \sqcup^a, \sqcap^a, \neg^a \rangle$ is a complete boolean algebra.*

8.7 Abstractions

The set $\mathbb{A} = \mathbb{A}^+ \cup \mathbb{A}^-$ of abstraction indexes covers positive/monotone abstractions $\mathbb{A}^+$ and negative/antitone abstractions $\mathbb{A}^-$. For each abstraction α^a, $a \in \mathbb{A}$, the following hypotheses **(27⁻)** or **(27⁻)** are needed in order to apply one of the Propositions (14) or its dual, (15) and/or (16) to fixpoints:

Hypothesis (27) [abstractions]

$$\forall a \in \mathbb{A}^+ : \alpha^a \in \langle \mathbb{L}, \leq^a \rangle \xmapsto{\bot,con} \langle \mathbb{M}^a, \sqsubseteq^a \rangle \qquad \mathbf{(27^+)}$$

$$\forall a \in \mathbb{A}^- : \alpha^a \in \langle \mathbb{L}, \leq^a \rangle \xmapsto{\top,co\text{-}con} \langle \mathbb{M}^a, \sqsubseteq^a \rangle \qquad \mathbf{(27^-)}$$

8.8 Abstract interpretation of primitive operators

Since different arguments of an operator $\Psi_n \in \mathfrak{O}^n$ may be abstracted differently, we assign to each operator $\Psi_n \in \mathfrak{O}^n$ a *signature* $Sig(\!|\Psi_n|\!)$ $\subseteq \{[a_1, \ldots, a_n]\to a_{n+1} \mid \forall i \in [1, n+1] : a_i \in \mathbb{A}\}$ specifying which abstraction is used for each argument and for the result. For constants $\Psi_0 \in \mathfrak{O}^0$, we write a as a shorthand for $[]\to a$. The abstract interpretation $\mathcal{I}[\![\Psi_n]\!]^{[a_1,\ldots,a_n]\to a}$ of an n-ary operator $\Psi_n \in \mathfrak{O}^n$ belongs to $(\mathbb{M}^{a_1} \times \ldots \times \mathbb{M}^{a_n}) \mapsto \mathbb{M}^a$ and satisfies the monotonicity condition (recall that $\sqsubseteq^+ \triangleq \sqsubseteq$ and $\sqsubseteq^- \triangleq \sqsupseteq$):

Hypothesis (28) [monotonicity of the abstract operators] *If $Sig(\!|\Psi_n|\!) = [a_1, \ldots, a_n]\to a \triangleq s$ then for all $x_1, y_1 \in \mathbb{M}^{a_1}, \ldots, x_n, y_n \in \mathbb{M}^{a_n}$:*

$$\forall i \in [1, n] : x_i \sqsubseteq^{a_i} y_i \Rightarrow$$
$$\mathcal{I}[\![\Psi_n]\!]^s(x_1, \ldots, x_n) \sqsubseteq^a \mathcal{I}[\![\Psi_n]\!]^s(y_1, \ldots, y_n)$$

By reflexivity, this hypothesis always holds for the abstract interpretation $\mathcal{I}[\![\Psi_n]\!]^a$ of a constant $\Psi_0 \in \mathfrak{O}^0$ with signature $Sig(\!|\Psi_0|\!) = a$. $\mathcal{I}[\![\Psi_n]\!]^a$ belongs to the abstract domain $\mathbb{M}^a$ isomorphic to $\mathbb{M}^{a^0} \mapsto \mathbb{M}^a = \emptyset \mapsto \mathbb{M}^a$.

8.9 Abstract environments

Abstract environments assign abstract values to free variables of formulae. So the set of abstract environments is $\mathbb{E}^A \triangleq \mathbb{X} \mapsto \mathbb{M}^A$ where the join of the abstract domains is defined as $\mathbb{M}^A \triangleq \bigcup_{a\in\mathbb{A}} \mathbb{M}^a$. An *environment type* $t \in \mathbb{X} \mapsto \mathbb{A}$ specifies which abstraction $\alpha^{t(X)} \in \mathbb{A}$ is used for each variable $X \in \mathbb{X}$. We define the *t-typed abstract environment domain* as $\mathbb{E}^t \triangleq \{\rho \in \mathbb{E}^A \mid \forall X \in \mathbb{X} : \rho(X) \in \mathbb{M}^{t(X)}\}$.

8.10 Abstract semantics

Given an abstract environment $\rho \in \mathbb{E}^A$, the *abstract semantics* of the language $\mathfrak{L}$ partially defines the semantic value $[\![\varphi]\!]^a\rho$ of formula $\varphi \in \mathfrak{L}$ in environment ρ for abstraction $\alpha^a \in \mathbb{A}$. The following definition of the semantic function $[\![\bullet]\!] \in \mathfrak{L} \mapsto (\mathbb{E} \mapsto \mathbb{L})$ is partial in that the least fixpoint $\mathrm{lfp}^{\leq^+}$ (or the greatest for $\mathrm{lfp}^{\leq^-}$) may not exist (this will be excluded by the well-definedness condition of Definition (23)) and values of variables may not be in the proper abstract domain (this will be excluded by the typing condition given in Definition (30) below):

Definition (29) [abstract semantics]

$$[\![\Psi_n(\varphi_1, \ldots, \varphi_n)]\!]^a\rho \triangleq \mathcal{I}[\![\Psi_n]\!]^{[a_1,\ldots,a_n]\to a}([\![\varphi_1]\!]^{a_1}\rho, \ldots, [\![\varphi_n]\!]^{a_n}\rho)$$
$$[\![X]\!]^a\rho \triangleq \rho(X)$$
$$[\![\boldsymbol{f}^p\, X \cdot \varphi_1]\!]^a\rho \triangleq \mathrm{lfp}^{(\sqsubseteq^a)^p}\, \lambda x \cdot [\![\varphi_1]\!]^a\rho[X := x]$$

Definition (29) covers the special case of constants $\Psi_0 \in \mathfrak{O}^0$ such that $[\![\Psi_0]\!]^a\rho \triangleq \mathcal{I}[\![\Psi_0]\!]^a()$.

8.11 Abstract semantics typing

We define a *typing* of the abstract semantics in order to check that the free variables consistently belong to the proper abstract domain and that the abstraction is sound $Type^{\sqsubseteq^a}$ (and complete $Type^=$). A type $t \Rightarrow a \in (\mathbb{X} \mapsto \mathbb{A}) \times \mathbb{A}$ specifies that a formula φ can be abstracted with α^a when its free variables $X \in FV(\varphi)$ are abstracted with $\alpha^{t(X)}$.

Definition (30) [abstract semantics typing]

$$Type^{=/\sqsubseteq^a}(\!|\Psi_n(\varphi_1, \ldots, \varphi_n)|\!) \triangleq$$
$$\{t \Rightarrow a \mid \exists[a_1, \ldots, a_n]\to a \in Sig(\!|\Psi_n|\!) :$$
$$\forall i \in [1, n] : t \Rightarrow a_i \in Type^{=/\sqsubseteq^a}(\!|\varphi_i|\!)\}$$
$$Type^{=/\sqsubseteq^a}(\!|X|\!) \triangleq \{t \Rightarrow t(X) \mid t \in \mathbb{X} \mapsto \mathbb{A}\}$$
$$Type^{=}(\!|\boldsymbol{f}^p\, X \cdot \varphi_1|\!) \triangleq \{t \Rightarrow a \mid a \in \mathbb{A}^p \wedge$$
$$t[X := a] \Rightarrow a \in Type^{=}(\!|\varphi_1|\!)\}$$
$$Type^{\sqsubseteq^a}(\!|\boldsymbol{f}^p\, X \cdot \varphi_1|\!) \triangleq \{t \Rightarrow a \mid t[X := a] \Rightarrow a \in Type^{\sqsubseteq^a}(\!|\varphi_1|\!)\}$$

For constants $\Psi_0 \in \mathfrak{O}^0$, Definition (30) amounts to $Type^{=/\sqsubseteq^a}(\!|\Psi_0|\!) \triangleq \{t \Rightarrow a \mid t \in \mathbb{X} \mapsto \mathbb{A} \wedge a \in Sig(\!|\Psi_0|\!)\}$. Obviously $Type^{=}(\!|\varphi|\!) \subseteq Type^{\sqsubseteq^a}(\!|\varphi|\!)$.

8.12 Well-definedness of the abstract semantics

Lemma (31) *Let $\varphi \in \mathfrak{L}$ be a formula, $\eta \in \mathbb{X} \mapsto \{+, -\}$ such that $Wd(\!|\varphi|\!)\eta = \mathrm{tt}$ and $t \Rightarrow a \in Type^{\sqsubseteq^a}(\!|\varphi|\!)$. Let the free variables of φ be $FV(\varphi) = \{X_1, \ldots, X_m\}$. Then Hypothesis (28) implies that for all environments $\rho \in \mathbb{X} \mapsto \mathbb{M}^A$ such that $\forall X \in \mathbb{X} : \rho(X) \in \mathbb{M}^{t(X)}$, the map $\lambda x_1, \ldots, x_m \cdot [\![\varphi]\!]^a\rho[X_1 := x_1, \ldots, X_m := x_m]$ is well-defined in $(\mathbb{M}^{t(X_1)} \times \ldots \times \mathbb{M}^{t(X_m)}) \mapsto \mathbb{M}^a$, monotone in x_i if $\eta(X_i) = +$ and antitone in x_i if $\eta(X_i) = -$, as follows:*

$$\forall i \in [1, m] : x_i \left(\sqsubseteq^{t(X_i)}\right)^{\eta(X_i)} y_i \Rightarrow$$
$$[\![\varphi]\!]^a\rho[X_1 := x_1, \ldots, X_m := x_m] \sqsubseteq^a [\![\varphi]\!]^a\rho[X_1 := y_1, \ldots, X_m := y_m]$$

Observe that Lemma (25) directly follows from Lemma (31) with $\mathbb{A}^- = \emptyset$, $\mathbb{A}^+ = \{\bullet\}$, the Galois isomorphism $\mathbf{1}$ on $\langle \mathbb{L}, \leq \rangle$ and Hypothesis (21) in lieu of Hypothesis (28).

8.13 Soundness/completeness hypotheses on the abstract interpretation of the operators

An abstract semantics must be sound in that it is an approximation of the abstraction of the concrete semantics. We say that the abstraction is complete (exact, precise, faithful, etc. can also be found in the literature) if the abstract semantics equals the abstraction of the semantics. For example the rule of signs [9] is not complete since the sign abstraction of an expression may be different from the sign of the value of the expression. However the rule of signs is sound since the sign of the value of an expression can never be in contradiction with the sign derived by the rule of signs, if any.

The following (semi-)commutation hypothesis [9] of the concrete and abstract interpretations of the primitive operators ensures the soundness (=)/completeness[2] ($\sqsubseteq^a$) of the abstract semantics for the abstraction α^a, as will be shown in Section 8.15 by application of Propositions (14) or its dual, (15) and/or (16) to fixpoints. We use the shorthand $=/\sqsubseteq^a$ for the two different commutation hypothesis with $=$ and semi-commutation hypothesis with $\sqsubseteq^a$. Grouping together the two cases shorten later proofs. When their distinction is needed, we respectively refer to Hypothesis $(32)^=$ and Hypothesis $(32)^{\sqsubseteq^a}$.

Hypothesis (32) [(semi-)commutation]

$$\forall \Psi_n \in \mathcal{D}^n : \forall [a_1, \ldots, a_n] \rightarrow a \in Sig(\!|\Psi_n|\!) : \forall x_1, \ldots, x_n \in \mathbb{L} :$$
$$\alpha^a(\mathcal{I}\llbracket \Psi_n \rrbracket(x_1, \ldots, x_n)) =/\sqsubseteq^a \mathcal{I}\llbracket \Psi_n \rrbracket^{[a_1, \ldots, a_n] \rightarrow a}(\alpha^{a_1}(x_1), \ldots, \alpha^{a_n}(x_n))$$

Hypothesis (32) covers the special case of constants $\Psi_0 \in \mathcal{D}^0$ for which $\forall \Psi_0 \in \mathcal{D}^0 : \forall a \in Sig(\!|\Psi_0|\!) : \alpha^a(\mathcal{I}\llbracket \Psi_0 \rrbracket) =/\sqsubseteq^a \mathcal{I}\llbracket \Psi_0 \rrbracket^a$.

8.14 Environment abstraction

Definition (33) [environment abstraction] *The abstraction of an environment $\rho \in \mathbb{E}$ of type $t \in \mathbb{X} \mapsto \mathbb{A}$ is $\dot{\alpha}^t(\rho) \in \mathbb{E}^t$ defined as:*

$$\dot{\alpha}^t(\rho) \triangleq \lambda X \cdot \alpha^{t(X)}(\rho(X)).$$

Lemma (34)

$$\dot{\alpha}^{t'}(\rho[X := x]) = \dot{\alpha}^{t'}(\rho)[X := \alpha^{t'(X)}(x)]$$

PROOF

$\dot{\alpha}^{t'}(\rho[X := x])$

$= \lambda Y \cdot \alpha^{t'(Y)}(\rho[X := x](Y))$ ⟅by Definition (33) of $\dot{\alpha}^{t'}(\rho)$⟆

$=$ ⟅by definition of substitution and conditional⟆
$\lambda Y \cdot (Y = X \; ? \; \alpha^{t'(X)}(x) \; ¿ \; \alpha^{t'(Y)}(\rho(Y)))$

$=$ ⟅by definition of substitution⟆
$(\lambda Y \cdot \alpha^{t'(Y)}(\rho(Y)))[X := \alpha^{t'(X)}(x)]$

$= \dot{\alpha}^{t'}(\rho)[X := \alpha^{t'(X)}(x)]$ ⟅by Definition (33) of $\dot{\alpha}^{t'}(\rho)$⟆ ∎

Lemma (35)

$$(\dot{\alpha}^{t[X := a]}(\rho))[X := y] = \dot{\alpha}^t(\rho)[X := y]$$

PROOF

$(\dot{\alpha}^{t[X := a]}(\rho))[X := y]$

$=$ ⟅by definition of substitution⟆
$\lambda Y \cdot (Y = X \; ? \; y \; ¿ \; \dot{\alpha}^{t[X := a]}(\rho)(Y))$

$=$ ⟅by Definition (33) of $\dot{\alpha}^t(\rho)$⟆
$\lambda Y \cdot (Y = X \; ? \; y \; ¿ \; \alpha^{t[X := a](Y)}(\rho(Y)))$

$=$ ⟅by definition of substitution when $X \neq Y$⟆
$\lambda Y \cdot (Y = X \; ? \; y \; ¿ \; \alpha^{t(Y)}(\rho(Y)))$

$= \lambda Y \cdot (Y = X \; ? \; y \; ¿ \; \dot{\alpha}^t(\rho)(Y))$ ⟅by Definition (33) of $\dot{\alpha}^t(\rho)$⟆

$= \dot{\alpha}^t(\rho)[X := y]$ ⟅by definition of substitution⟆ ∎

[2] We should say *relative completeness* to stress the fact that we reason in set theoretical terms, so that, in logical terms, an oracle is assumed to exist for logical implication.

8.15 Soundness and completeness of the abstract semantics

We start with a lemma in order to extend the (semi-)commutation condition of Hypothesis (32) from operators to the full abstract semantics.

Lemma (36) *For all well-defined formulae $\varphi \in \mathcal{L}$ with $Wd(\!|\varphi|\!)\eta = \mathtt{tt}$, for all $t \in \mathbb{X} \mapsto \mathbb{A}$ and $a \in \mathbb{A}$ such that $t \Rightarrow a \in Type^{=/\sqsubseteq^a}(\!|\varphi|\!)$, for all $\rho \in \mathbb{E}$ we have:*

$$\alpha^a(\llbracket \varphi \rrbracket \rho) =/\sqsubseteq^a \llbracket \varphi \rrbracket^a(\dot{\alpha}^t(\rho)). \qquad \textbf{(37)}$$

PROOF The proof is by structural induction on the formula φ:

— For variables $\varphi = X \in \mathbb{X}$:

$\alpha^a(\llbracket X \rrbracket \rho)$

$= \alpha^a(\rho(X))$ ⟅by Definition (22)⟆

$=$ ⟅$t \Rightarrow a \in Type^{=/\sqsubseteq^a}(\!|X|\!)$ implies that $t(X) = a$ by Definition (30)⟆
$\alpha^{t(X)}(\rho(X))$

$= \dot{\alpha}^t(\rho)(X)$ ⟅by Definition (33)⟆

$= \llbracket X \rrbracket^a(\dot{\alpha}^t)$ ⟅by Definition (29)⟆

— For n-ary operators $\varphi = \Psi_n(\varphi_1, \ldots, \varphi_n)$:

$\alpha^a(\llbracket \Psi_n(\varphi_1, \ldots, \varphi_n) \rrbracket \rho)$

$= \alpha^a(\mathcal{I}\llbracket \Psi_n \rrbracket(\llbracket \varphi_1 \rrbracket \rho, \ldots, \llbracket \varphi_n \rrbracket \rho))$ ⟅by Definition (22)⟆

$=/\sqsubseteq^a$ ⟅by Hypothesis (32)⟆
$\mathcal{I}\llbracket \Psi_n \rrbracket^{[a_1, \ldots, a_n] \rightarrow a}(\alpha^{a_1}(\llbracket \varphi_1 \rrbracket \rho), \ldots, \alpha^{a_n}(\llbracket \varphi_n \rrbracket \rho))$

$=/\sqsubseteq^a$ ⟅by Hypothesis (28) in the $\sqsubseteq^a$ case and structural induction Hypothesis (37) applied to $\varphi_1, \ldots, \varphi_n$⟆
$\mathcal{I}\llbracket \Psi_n \rrbracket^{[a_1, \ldots, a_n] \rightarrow a}(\llbracket \varphi_1 \rrbracket^{a_1}(\dot{\alpha}^t(\rho)), \ldots, \llbracket \varphi_n \rrbracket^{a_n}(\dot{\alpha}^t(\rho)))$

$= \llbracket \Psi_n(\varphi_1, \ldots, \varphi_n) \rrbracket^a(\dot{\alpha}^t(\rho))$ ⟅by Definition (29)⟆

— For fixpoints $\varphi = \mathbf{f}^p\, X \cdot \varphi_1$, we define:

$$\mathcal{F} \triangleq \lambda x \cdot \llbracket \varphi_1 \rrbracket \rho[X := x] \qquad \mathcal{G} \triangleq \lambda y \cdot \llbracket \varphi_1 \rrbracket^a \dot{\alpha}^t(\rho)[X := y]$$

— We have $Wd(\!|\mathbf{f}^p\, X \cdot \varphi_1|\!)\eta = \mathtt{tt}$ hence by Definition (23), $Wd(\!|\varphi_1|\!)\eta[X := +] = \mathtt{tt}$ which together with $\eta[X := +](X) = +$ and according to Lemma (25) implies that $\mathcal{F} \in \mathbb{L} \mapsto \mathbb{L}$ is well-defined and $\leq$-monotone.

— Moreover $t \Rightarrow a \in Type^{=/\sqsubseteq^a}(\!|\mathbf{f}^p\, X \cdot \varphi_1|\!)$ so that by Definition (30), $t[X := a] \Rightarrow a \in Type^{=/\sqsubseteq^a}(\!|\varphi_1|\!)$. It follows by Lemma (31) that $\mathcal{G}$ is well-defined in $\mathbb{M}^{t[X := a](X)} \mapsto \mathbb{M}^a = \mathbb{M}^a \mapsto \mathbb{M}^a$ since $t[X := a](X) = a$. Moreover $\eta(X) = +$ so $\left(\sqsubseteq^{t[X := a](X)}\right)^{\eta(X)} = (\sqsubseteq^a)^+ = \sqsubseteq^a$ so that Lemma (31) implies that $\mathcal{G}$ is $\sqsubseteq^a$-monotone.

— For the (semi-)commutation condition, we have:

$\alpha^a(\mathcal{F}(x))$

$= \alpha^a(\llbracket \varphi_1 \rrbracket \rho[X := x])$ ⟅by definition of $\mathcal{F}(x)$⟆

$=/\sqsubseteq^a$ ⟅by induction hypothesis (37) applied to φ_1, which by Definition (30) has type $t[X := a] \Rightarrow a$⟆
$\llbracket \varphi_1 \rrbracket^a(\dot{\alpha}^{t[X := a]}(\rho[X := x]))$

$=$ ⟅by Lemma (34) with $t' = t[X := a]$ so that $t'(X) = a$⟆
$\llbracket \varphi_1 \rrbracket^a(\dot{\alpha}^{t[X := a]}(\rho)[X := \alpha^a(x)]))$

$= \llbracket \varphi_1 \rrbracket^a \dot{\alpha}^t(\rho)[X := \alpha^a(x)]$ ⟅by Lemma (35)⟆

$= \mathcal{G}(\alpha^a(x))$ ⟅by definition of $\mathcal{G}$⟆

— We now prove that: $\alpha^a(\mathrm{lfp}^{\leq^p} \mathcal{F}) =/\sqsubseteq^a \mathrm{lfp}^{(\sqsubseteq^a)^p} \mathcal{G}$ **(38)**

- In the equality case $(38)^=$, we have $t \Rightarrow a \in \mathit{Type}^=(\!|\mathbf{f}^p X \cdot \varphi_1|\!)$ so that by Definition (30), $a \in \mathbb{A}^p$. If p is $+$ then α^a is strict and continuous by Hypothesis (27) and we conclude by Proposition (14) that $(38)^=$ holds. If p is $-$ then α^a is co-strict and co-continuous by Hypothesis (27) and we conclude by the dual of Proposition (14) that $(38)^=$ holds.
- In the inequality case $(38)^{\sqsubseteq^a}$, α^a is monotone in both cases $a \in \mathbb{A}^+$ and $a \in \mathbb{A}^+$ by Hypothesis (27). So if p is $+$, we conclude that $(38)^{\sqsubseteq^a}$ holds by Proposition (15) whereas when p is $-$, we conclude that $(38)^{\sqsubseteq^a}$ holds by Proposition (16). Q.E.D.

— Finally we get:

$$\begin{array}{ll}
\alpha^a([\![\mathbf{f}^p X \cdot \varphi_1]\!]\rho) & \\
= \alpha^a(\mathrm{lfp}^{\leq^p} \lambda x \cdot [\![\varphi_1]\!]\rho[X := x]) & \wr\text{by Definition (22)}\wr \\
= \alpha^a(\mathrm{lfp}^{\leq^p} \mathcal{F}) & \wr\text{by definition of } \mathcal{F}\wr \\
=/\sqsubseteq^a \mathrm{lfp}^{(\sqsubseteq^a)^p} \mathcal{G} & \wr\text{by (38)}\wr \\
= \mathrm{lfp}^{(\sqsubseteq^a)^p} \lambda y \cdot [\![\varphi_1]\!]^a(\dot{\alpha}^t(\rho))[X := y] & \wr\text{by definition of } \mathcal{G}\wr \\
= [\![\mathbf{f}^p X \cdot \varphi_1]\!]^a(\dot{\alpha}^t(\rho)) & \wr\text{by Definition (29)}\wr \ \blacksquare
\end{array}$$

In Section 10, we will have $\mathbb{A}^+ = \mathbb{A}^- = \mathbb{A}$ and stronger properties than Hypothesis (27), as follows:

Hypothesis (39) [Galois abstraction]

$$\langle \mathbb{L}, \leq^a\rangle \underset{\alpha^a}{\overset{\gamma^a}{\leftrightarrows}} \langle \mathbb{M}^a, \sqsubseteq^a\rangle \ \wedge\ \alpha^a \in \langle \mathbb{L}, \leq^a\rangle \xmapsto{\top\text{,co-con}} \langle \mathbb{M}^a, \sqsubseteq^a\rangle \quad \mathbf{(39^+)}$$

or

$$\langle \mathbb{L}, \geq^a\rangle \underset{\alpha^a}{\overset{\gamma^a}{\leftrightarrows}} \langle \mathbb{M}^a, \sqsupseteq^a\rangle \ \wedge\ \alpha^a \in \langle \mathbb{L}, \leq^a\rangle \xmapsto{\bot\text{,con}} \langle \mathbb{M}^a, \sqsubseteq^a\rangle \quad \mathbf{(39^-)}$$

In case **(39⁺)**, α^a is a complete join morphism whence strict and continuous, so that Hypothesis (27) is satisfied, with $a \in \mathbb{A}^+$. In case **(39⁻)**, α^a is a complete meet morphism whence co-strict and co-continuous, so that Hypothesis (27) is satisfied, with $a \in \mathbb{A}^-$.

For environments, we define:

$$\dot{\gamma}^t(\rho) \triangleq \lambda X \cdot \gamma^{t(X)}(\rho(X)).$$

For semantics, we define:

$$\alpha^{t \Rightarrow a} \triangleq \lambda \phi \cdot \alpha^a \circ \phi \circ \dot{\gamma}^t \quad \text{and} \quad \gamma^{t \Rightarrow a} \triangleq \lambda \psi \cdot \gamma^a \circ \psi \circ \dot{\alpha}^t.$$

Theorem (40) *Hypothesis* (39) *implies that for all well-defined formulae* $\varphi \in \mathfrak{L}$ *of type* $t \Rightarrow a \in \mathit{Type}^{=/\sqsubseteq^a}(\!|\varphi|\!)$, *we have:*

$$\alpha^{t \Rightarrow a}([\![\varphi]\!]) =/\sqsubseteq^a [\![\varphi]\!]^a.$$

PROOF If $t \in \mathbb{X} \mapsto \mathbb{A}$ and $p \in \{+, -\}$, Hypothesis **(39ᵖ)** implies $\langle \mathbb{L}, (\leq^a)^p\rangle \underset{\alpha^a}{\overset{\gamma^a}{\leftrightarrows}} \langle \mathbb{M}^a, (\sqsubseteq^a)^p\rangle$ whence

$$\langle \mathbb{E}, (\dot{\leq}^a)^p\rangle \underset{\dot{\alpha}^t}{\overset{\dot{\gamma}^t}{\leftrightarrows}} \langle \mathbb{E}^t, (\dot{\sqsubseteq}^a)^p\rangle \quad \mathbf{(41)}$$

and $\langle \mathbb{E} \mapsto \mathbb{M}, (\dot{\leq}^a)^p\rangle \underset{\dot{\alpha}^{t \Rightarrow a}}{\overset{\dot{\gamma}^{t \Rightarrow a}}{\leftrightarrows}} \langle \mathbb{E}^t \mapsto \mathbb{M}^a, (\dot{\sqsubseteq}^a)^p\rangle$ so that for all $a \in \mathbb{A}$ and $\rho \in \mathbb{E}^t$, we have:

$$\begin{array}{lll}
\alpha^{t \Rightarrow a}([\![\varphi]\!])(\rho) & = \alpha^a([\![\varphi]\!](\dot{\gamma}^t(\rho))) & \wr\text{by definition of } \alpha^{t \Rightarrow a}\wr \\
=/\sqsubseteq^a [\![\varphi]\!]^a(\dot{\alpha}^t(\dot{\gamma}^t(\rho))) & & \wr\text{by Lemma (36)}\wr \\
= [\![\varphi]\!]^a \rho & \wr\text{by } \dot{\alpha}^t \circ \dot{\gamma}^t = \mathbf{1} \text{ in the Galois surjection (41)}\wr & \blacksquare
\end{array}$$

9. State-based abstraction for model-checking

One way of understanding model-checking is to define a *model-checking specification language* $\mathfrak{L}$ (chosen as a subset of the $\widehat{\mu}^\star$-calculus) and then to formally derive *the specification of the model-checking algorithms* by a sound and complete abstraction of the trace-based semantics of $\mathfrak{L}$. This is illustrated in this Section 9 on the *propositional μ-calculus* [18, 19]:

$$\begin{array}{l}
\varphi ::= \sigma_S \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \wedge \varphi_2 \mid \neg \varphi_1 \mid [\!]\, \varphi_1 \mid \langle\!\rangle\, \varphi_1 \mid \\
\quad X \mid \mu X \cdot \varphi_1 \mid \nu X \cdot \varphi_1
\end{array}$$

where, given a transition system $\langle \mathbb{S}, \tau\rangle$, the modal operators $[\!]$ and $\langle\!\rangle$ are defined as follows:

$$[\!]\, \varphi_1 \triangleq \exists \pi_\tau : \oplus \varphi_1 \qquad \langle\!\rangle\, \varphi_1 \triangleq \forall \pi_\tau : \oplus \varphi_1 \quad \mathbf{(42)}$$

The *current state abstraction*:

$$\alpha^\bullet(M) \triangleq \{\sigma_i \mid \langle i, \sigma\rangle \in M\} \quad \gamma^\bullet(S) \triangleq \{\langle i, \sigma\rangle \mid \sigma_i \in S\} = \sigma\{\!|S|\!\}$$

is self-dual (i.e. $\neg\alpha^\bullet(\neg M) = \alpha^\bullet(M)$ and $\neg\gamma^\bullet(\neg S) = \gamma^\bullet(S)$) so that we have the following dual Galois connections:

$$\langle \mathbb{M}, \subseteq\rangle \underset{\alpha^\bullet}{\overset{\gamma^\bullet}{\leftrightarrows}} \langle \wp(\mathbb{S}), \subseteq\rangle \quad \text{and} \quad \langle \mathbb{M}, \supseteq\rangle \underset{\alpha^\bullet}{\overset{\gamma^\bullet}{\leftrightarrows}} \langle \wp(\mathbb{S}), \supseteq\rangle.$$

It follows that $\alpha^\bullet$ is both a complete join and meet morphism which implies Hypothesis (39) (whence (27)) and the commutation Hypothesis $(32)^=$ for $\vee$ and $\wedge$. For $\neg$, $\alpha^\bullet([\![\neg\varphi]\!]\rho) = \alpha^\bullet(\neg[\![\varphi]\!]\rho) = \neg\alpha^\bullet([\![\varphi]\!]\rho)$ by self-duality. The two remaining cases are $\alpha^\bullet([\![[\!]\,\varphi]\!]\rho) = \widetilde{pre}[\tau](\alpha^\bullet([\![\varphi]\!]\rho))$ and $\alpha^\bullet([\![\langle\!\rangle\,\varphi]\!]\rho) = pre[\tau](\alpha^\bullet([\![\varphi]\!]\rho))$, where:

$$\begin{array}{rll}
post[t](P) & \triangleq \{s' \mid \exists s : t(s, s') \wedge s \in P\} & \\
\widetilde{post}[t](P) & \triangleq \neg post[t](\neg P) = \{s' \mid \forall s : t(s, s') \Rightarrow s \in P\} & \mathbf{(43)} \\
pre[t](P) & \triangleq post[t^{-1}](P) = \{s \mid \exists s' : t(s, s') \wedge s' \in P\} & \\
\widetilde{pre}[t](P) & \triangleq \neg pre[t](\neg P) = \{s \mid \forall s' : t(s, s') \Rightarrow s' \in P\} & \mathbf{(44)}
\end{array}$$

Observe that $\alpha^\bullet(\curvearrowleft\{\!|X|\!\}) = \alpha^\bullet(X)$ so that reversal is of no real interest in the propositional μ-calculus.

For any propositional μ-calculus formula φ, Theorem (40) implies $\alpha^\bullet([\![\varphi]\!]) = [\![\varphi]\!]^\bullet$, where the abstract semantics of Definition (29) is as follows:

$$\begin{array}{ll}
[\![\varphi_1 \vee \varphi_2]\!]^\bullet\rho = [\![\varphi_1]\!]^\bullet\rho \cup [\![\varphi_2]\!]^\bullet\rho & [\![\sigma_S]\!]^\bullet\rho = S \\
[\![\varphi_1 \wedge \varphi_2]\!]^\bullet\rho = [\![\varphi_1]\!]^\bullet\rho \cap [\![\varphi_2]\!]^\bullet\rho & [\![\neg\varphi_1]\!]^\bullet\rho = \neg[\![\varphi_1]\!]^\bullet\rho \\
[\![[\!]\,\varphi_1]\!]^\bullet\rho = \widetilde{pre}[\tau]([\![\varphi]\!]^\bullet\rho) & [\![\langle\!\rangle\,\varphi_1]\!]^\bullet\rho = pre[\tau]([\![\varphi]\!]^\bullet\rho) \\
[\![\mu X \cdot \varphi_1]\!]^\bullet\rho = \mathrm{lfp}^{\subseteq} \lambda x \cdot [\![\varphi_1]\!]^\bullet\rho[X := x] & [\![X]\!]^\bullet\rho = \rho(X) \\
[\![\nu X \cdot \varphi_1]\!]^\bullet\rho = \mathrm{gfp}^{\subseteq} \lambda x \cdot [\![\varphi_1]\!]^\bullet\rho[X := x] &
\end{array}$$

The classical boolean specification of the model-checking algorithms [19] follows by application of the characteristic predicate isomorphism $\langle \wp(\mathbb{S}), \subseteq\rangle \underset{\theta}{\overset{\theta^{-1}}{\leftrightarrows}} \langle \mathbb{S} \mapsto \mathbb{B}, \dot{\Rightarrow}\rangle$ with $\theta(S) \triangleq \lambda s \cdot s \in S$ and the finiteness hypothesis of $\mathbb{S}$.

We think that this completeness result corresponds to the intuitive understanding that set-based model-checking algorithms are an economical but equivalent way of reasoning on trace-based specifications. In our opinion, this is misleading since an incomplete abstraction is hidden in the $\forall$- and $\exists$-based definitions (42). In the following sections we study these abstractions explicitly in order to exhibit the source of incompleteness.

10. The checking abstractions

Checking abstractions check a specification $\phi \in \mathbb{M}$ for a model $M \in \mathbb{M}$ (e.g. generated by a transition system). They provide set of present states s such that all traces in M (for the universal abstraction) or some trace in M (for the existential one) with present state s do satisfy ϕ. This is an approximation since a set of computation traces is approximated by a set of present states.

10.1 The universal checking abstraction

The universal checking abstraction $\alpha^\forall_M(\phi)$ checks for each state $s \in \mathbb{S}$ that all traces of the model $M \in \mathbb{M}$ which current state is s definitely satisfy the specification $\phi \in \mathbb{M}$

Definition (45) [universal checking abstraction]

$$\alpha^\forall_M(\phi) \triangleq \{s \mid M_{\downarrow s} \subseteq \phi\} \qquad \text{universal abstraction} \quad \textbf{(46)}$$

$$\gamma^\forall_M(S) \triangleq \{\langle i, \sigma\rangle \mid \langle i, \sigma\rangle \in M \wedge \sigma_i \in S\} \quad \text{u. concretization} \quad \textbf{(47)}$$

We have $\langle \mathbb{M}, \supseteq\rangle \underset{\alpha^\forall_M}{\overset{\gamma^\forall_M}{\leftrightarrows}} \langle \wp(\mathbb{S}), \supseteq\rangle$. Moreover by the totality Lemma (12), $\gamma^\forall_M$ is injective, so $\langle \mathbb{M}, \supseteq\rangle \underset{\alpha^\forall_M}{\overset{\gamma^\forall_M}{\leftrightarrows\!\!\!\!\rightarrow}} \langle \wp(\mathbb{S}), \supseteq\rangle$ whence $\alpha^\forall_M$ is $\emptyset$-strict. Finally, $\alpha^\forall_M$ is $\subseteq$-continuous: $\alpha^\forall_M \in \langle \mathbb{M}, \subseteq\rangle \overset{\text{con}}{\longmapsto} \langle \wp(\mathbb{S}), \subseteq\rangle$. Otherwise stated the universal abstraction satisfies Hypothesis **(39$^-$)**:

$$\langle \mathbb{M}, \supseteq\rangle \underset{\alpha^\forall_M}{\overset{\gamma^\forall_M}{\leftrightarrows\!\!\!\!\rightarrow}} \langle \wp(\mathbb{S}), \supseteq\rangle,\ \alpha^\forall_M \in \langle \mathbb{M}, \subseteq\rangle \overset{\perp,\text{con}}{\longmapsto} \langle \wp(\mathbb{S}), \subseteq\rangle \quad \textbf{(48)}$$

It follows that in order to apply Theorem (40), we just have to design an abstract semantics satisfying the (semi-)commutation condition Hypothesis (32). We respectively write $\dot{\alpha}^\forall_M$ and $\vec{\alpha}^\forall_M$ for the pointwise extensions of $\alpha^\forall_M$ to $\mathbb{X} \mapsto \mathbb{M}$ and $(\mathbb{X} \mapsto \mathbb{M}) \mapsto \mathbb{M}$.

10.2 The existential checking abstraction

The existential checking abstraction $\alpha^\forall_M(\phi)$ checks for each state $s \in \mathbb{S}$ that at least one trace of the model $M \in \mathbb{M}$ which current state is s does satisfy the specification $\phi \in \mathbb{M}$.

Definition (49) [existential checking abstraction]

$$\begin{aligned}\alpha^\exists_M(\phi) &\triangleq \neg\alpha^\forall_M(\neg\phi) && \text{existential abstraction}\\ &= \{s \mid (M_{\downarrow s} \cap \phi) \neq \emptyset\}\\ \gamma^\exists_M(\phi) &\triangleq \neg\gamma^\forall_M(\neg\phi) && \text{existential concretization}\\ &= \{\langle i, \sigma\rangle \mid (\langle i, \sigma\rangle \in M) \Rightarrow (\sigma_i \in S)\}\end{aligned}$$

so that, by duality, the existential abstraction satisfies Hypothesis **(39$^+$)**:

$$\langle \mathbb{M}, \subseteq\rangle \underset{\alpha^\exists_M}{\overset{\gamma^\exists_M}{\leftrightarrows\!\!\!\!\rightarrow}} \langle \wp(\mathbb{S}), \subseteq\rangle,\ \alpha^\exists_\in \langle \mathbb{M}, \subseteq\rangle \overset{\top,\text{co-con}}{\longmapsto} \langle \wp(\mathbb{S}), \subseteq\rangle \qquad \textbf{(50)}$$

Again, in order to apply Theorem (40), we just have to design an abstract semantics satisfying the (semi-)commutation condition Hypothesis (32). We respectively write $\dot{\gamma}^\forall_M$ and $\vec{\gamma}^\forall_M$ for the pointwise extensions of $\gamma^\forall_M$ to $\mathbb{X} \mapsto \mathbb{M}$ and $(\mathbb{X} \mapsto \mathbb{M}) \mapsto \mathbb{M}$.

11. The transition system checking abstraction of temporal calculi/logics

We now derive the classical interpretation/semantics of temporal calculi/logics (where a formula φ is interpreted as a set of states in which φ is true) by abstract interpretation of the trace-based semantics of Section 5 using the checking abstractions of Section 10 for the model $\mathcal{M}_\tau$ generated by a total transition system $\langle \mathbb{S}, \tau\rangle$. Hypothesis (32) leads to a calculational design of the abstract semantics $[\![\varphi]\!]^\forall_\tau =/\subseteq \vec{\alpha}^\forall_{\mathcal{M}_\tau}([\![\varphi]\!])$ and $[\![\varphi]\!]^\exists_\tau =/\subseteq \vec{\alpha}^\exists_{\mathcal{M}_\tau}([\![\varphi]\!])$. The sources of incompleteness, even for finite systems, are clearly identified thus showing that classical model checking is <u>not</u> complete with respect to trace-based semantics (whereas it is <u>after</u> abstraction to sets of states, as shown in Section 9).

11.1 Abstraction of state and transition models

We study typical cases, starting with $\boldsymbol{\sigma}_S$:

$$\begin{aligned}&\alpha^\forall_{\mathcal{M}_\tau}(\boldsymbol{\sigma}\{\!|S|\!\})\\ =\ &\{s \mid \mathcal{M}_{\tau\downarrow s} \subseteq \boldsymbol{\sigma}\{\!|S|\!\}\} && \wr\text{by Definition (45)}\int\\ =\ &\quad\wr\text{by Definitions (4) and (1)}\int\\ &\{s \mid \{\langle i, \sigma\rangle \in \mathcal{M}_\tau \mid \sigma_i = s\} \subseteq \{\langle i, \sigma\rangle \in \mathbb{T} \mid \sigma_i \in S\}\}\\ =\ &\{s \mid \forall\langle i, \sigma\rangle \in \mathcal{M}_\tau : (\sigma_i = s) \Rightarrow (\sigma_i \in S)\} && \wr\text{by def. of} \subseteq\int\\ =\ &\quad\wr\text{by Hypothesis (10) implying Lemma (12) so that } \forall s :\\ &\qquad \exists\langle i, \sigma\rangle \in \mathcal{M}_\tau : \sigma_i = s\int\\ =\ &S\\ =\ &[\![\boldsymbol{\sigma}_S]\!]^\forall_\tau(\dot{\alpha}^\forall_{\mathcal{M}_\tau}(\rho)) && \wr\text{by defining } [\![\boldsymbol{\sigma}_S]\!]^\forall_\tau\rho \triangleq S\int\end{aligned}$$

The same way, we get: $[\![\boldsymbol{\sigma}_S]\!]^\exists_\tau\rho \triangleq S$, $[\![\boldsymbol{\pi}_\tau]\!]^\forall_\tau\rho \triangleq \{s \mid \forall s' : t(s, s') \Rightarrow \tau(s, s')\}$ and $[\![\boldsymbol{\pi}_\tau]\!]^\exists_\tau\rho \triangleq \{s \mid \exists s' : t(s, s') \wedge \tau(s, s')\}$.

11.2 Semi-commutation for the predecessor operator

For the predecessor operator $\oplus$, we show that:

$$\alpha^\forall_{\mathcal{M}_\tau}(\oplus\{\!|\mathcal{X}|\!\}) \supseteq/\subseteq_{FD(\mathcal{X})} \widetilde{pre}[\tau](\alpha^\forall_{\mathcal{M}_\tau}(\mathcal{X})) \qquad \textbf{(51)}$$

($\supseteq/\subseteq_{FD(\mathcal{X})}$ denotes that fact that the inclusion $\supseteq$ always holds whereas $\subseteq$ holds whenever the temporal model $\mathcal{X}$ is forward closed, that is $\mathcal{X} = Fd(\mathcal{X})$ (written for short $FD(\mathcal{X})$).) Proof Given $\mathcal{M}_\tau$ generated by the transition system $\langle \mathbb{S}, \tau\rangle$, we define:

$$\begin{aligned}A(s) \triangleq\ &\forall i : \forall\sigma : (\sigma_i = s \wedge \tau(s, \sigma_{i+1}) \wedge && \textbf{(52)}\\ &\quad \langle i+1, \sigma\rangle \in \mathcal{M}_\tau) \Rightarrow (\langle i+1, \sigma\rangle \in \mathcal{X})\\ B(s) \triangleq\ &\forall s' : \forall j : \forall\sigma' : (\tau(s, s') \wedge \sigma'_{j+1} = s' \wedge && \textbf{(53)}\\ &\quad \langle j+1, \sigma'\rangle \in \mathcal{M}_\tau) \Rightarrow (\langle j+1, \sigma'\rangle \in \mathcal{X})\end{aligned}$$

We first prove that $\forall s : B(s) \Rightarrow A(s)$: **(54)**

$$\begin{aligned}&B(s) \wedge (\sigma_i = s \wedge \tau(s, \sigma_{i+1}) \wedge \langle i+1, \sigma\rangle \in \mathcal{M}_\tau)\\ \Rightarrow\ &\quad\wr B(s) \text{ with } s' = \sigma_{i+1},\ j = i,\ \sigma' = \sigma \text{ so that } \tau(s, s') \wedge\\ &\quad \sigma_{j+1} = s' \wedge \langle j+1, \sigma'\rangle \in \mathcal{M}_\tau \text{ implies } \langle j+1, \sigma'\rangle =\\ &\quad \langle i+1, \sigma\rangle \in \mathcal{X}\int\\ \Rightarrow\ &\langle i+1, \sigma\rangle \in \mathcal{X} && \wr\text{proving } A(s)\int \quad \text{Q.E.D.}\end{aligned}$$

Next, we prove that $FD(\mathcal{X}) \Rightarrow (\forall s : A(s) \Rightarrow B(s))$: **(55)**

$A(s) \wedge \tau(s, s') \wedge \sigma'_{j+1} = s' \wedge \langle j+1, \sigma'\rangle \in \mathcal{M}_\tau$

$\Rightarrow$ {by Hypothesis (10) implying Lemma (12) so that there exists a trace $\sigma = \eta[_j s[_{j+1}\sigma'$ of $\mathcal{M}_\tau$. We have $s = \sigma_j \wedge \tau(s, \sigma_{j+1}) \wedge \langle j+1, \sigma\rangle \in \mathcal{M}_\tau$ so that $A(s)$ implies $\langle j+1, \sigma\rangle \in \mathcal{X}$}

$\langle j+1, \sigma\rangle \in \mathcal{X} \wedge \sigma = \eta[_j s[_{j+1}\sigma'$

$\Rightarrow$ {$FD(\mathcal{X})$ so that $\sigma = \eta[_j s[_{j+1}\sigma'$ and $\langle j+1, \sigma\rangle \in \mathcal{X}$ implies $\langle j+1, \sigma'[_{j+1}\sigma'\rangle = \langle j+1, \sigma'\rangle \in \mathcal{X}$}

$\langle j+1, \sigma'\rangle \in \mathcal{X}$ {proving $B(s)$} Q.E.D.

We can now consider the (semi-)commutation condition (51) for the predecessor operator $\oplus$:

$\alpha^\vee_{\mathcal{M}_\tau}(\oplus\llbracket\mathcal{X}\rrbracket)$

$= \alpha^\vee_{\mathcal{M}_\tau}(\{\langle i, \sigma\rangle \in \mathbb{T} \mid \langle i+1, \sigma\rangle \in \mathcal{X}\})$ {by Definition (2)}

$=$ {by Definition (4) of $\bullet\downarrow_\bullet$ and Definition (45) of $\alpha^\vee_{\mathcal{M}_\tau}$}

$\{s \mid \{\langle i, \sigma\rangle \in \mathcal{M}_\tau \mid \sigma_i = s\} \subseteq \{\langle i, \sigma\rangle \in \mathbb{T} \mid \langle i+1, \sigma\rangle \in \mathcal{X}\}\}$

$=$ {by Definition of $\subseteq$}

$\{s \mid \forall i : \forall \sigma : (\sigma_i = s \wedge \langle i, \sigma\rangle \in \mathcal{M}_\tau) \Rightarrow (\langle i+1, \sigma\rangle \in \mathcal{X})\}$

$=$ {by Definition (11) of $\mathcal{M}_\tau$ and Lemma (17) so that $\langle i, \sigma\rangle \in \mathcal{M}_\tau$ iff $\langle i+1, \sigma\rangle \in \mathcal{M}_\tau$}

$\{s \mid \forall i : \forall \sigma : (\sigma_i = s \wedge \tau(s, \sigma_{i+1}) \wedge \langle i+1, \sigma\rangle \in \mathcal{M}_\tau) \Rightarrow (\langle i+1, \sigma\rangle \in \mathcal{X})\}$

$= \{s \mid A(s)\}$ {by Definition (52) of A}

$\supseteq / \subseteq_{FD(\mathcal{X})}$ $\{s \mid B(s)\}$ {by Lemmata (54) and (55)}

$=$ {by Definition (53) of B where $i = j+1$}

$\{s \mid \forall s' : \forall i : \forall \sigma' : (\tau(s, s') \wedge \sigma'_i = s' \wedge \langle i, \sigma'\rangle \in \mathcal{M}_\tau) \Rightarrow (\langle i, \sigma'\rangle \in \mathcal{X})\}$

$=$ {by letting $\sigma = \sigma'$, Definition (4) of $\bullet\downarrow_\bullet$ and logical definition of $\Rightarrow$}

$\{s \mid \forall s' : \tau(s, s') \Rightarrow [\forall i : \forall \sigma : \langle i, \sigma\rangle \in \mathcal{M}_\tau\downarrow_{s'}) \Rightarrow (\langle i, \sigma\rangle \in \mathcal{X})]\}$

$=$ {by set-theoretical definition of $\subseteq$}

$\{s \mid \forall s' : \tau(s, s') \Rightarrow [s' \in \{s'' \mid \mathcal{M}_\tau\downarrow_{s''} \subseteq \mathcal{X}\}]\}$

$=$ {by Definition (46) of $\alpha^\vee_{\mathcal{M}_\tau}$}

$\{s \mid \forall s' : \tau(s, s') \Rightarrow s' \in \alpha^\vee_{\mathcal{M}_\tau}(\mathcal{X})\}$

$= \widetilde{pre}[\tau](\alpha^\vee_{\mathcal{M}_\tau}(\mathcal{X}))$ {by Definition (44) of $\widetilde{pre}$} ∎

In general, $\alpha^\vee_{\mathcal{M}_\tau}(\oplus\llbracket\mathcal{X}\rrbracket) \subsetneq \widetilde{pre}[\tau](\alpha^\vee_{\mathcal{M}_\tau}(\mathcal{X}))$ **(56)**

PROOF Consider the counter-example $\mathbb{S} \triangleq \{\circ, \bullet\}$, $\tau \triangleq \{\langle\bullet, \bullet\rangle, \langle\bullet, \circ\rangle, \langle\circ, \circ\rangle\}$ so that $\mathcal{M}_\tau = \{\langle i, \lambda\ell\cdot\bullet\rangle, \langle j, \lambda\ell\cdot\circ\rangle, \langle k, \lambda\ell\cdot(\ell < m\ ?\ \bullet\ ¿\ \circ)\rangle\}$. We let $\mathcal{X} \triangleq \{\langle i, \sigma\rangle \mid \forall j < i : \sigma_j = \bullet\}$ so that $\neg FD(\mathcal{X})$. We have $\mathcal{M}_\tau\downarrow_\bullet = \{\langle i, \lambda\ell\cdot\bullet\rangle, \langle k, \lambda\ell\cdot(\ell < m\ ?\ \bullet\ ¿\ \circ)\rangle \mid k < m\}$, $\mathcal{M}_\tau\downarrow_\circ = \{\langle j, \lambda\ell\cdot\circ\rangle, \langle k, \lambda\ell\cdot(\ell < m\ ?\ \bullet\ ¿\ \circ)\rangle \mid k \geq m\}$ and $\oplus\llbracket\mathcal{X}\rrbracket = \{\langle i, \sigma\rangle \mid \forall j \leq i : \sigma_j = \bullet\}$. We have $\alpha^\vee_{\mathcal{M}_\tau}(\oplus\llbracket\mathcal{X}\rrbracket) = \{s \mid \mathcal{M}_\tau\downarrow_s \subseteq \oplus\llbracket\mathcal{X}\rrbracket\} = \{\bullet\}$ whereas $\widetilde{pre}[\tau](\alpha^\vee_{\mathcal{M}_\tau}(\mathcal{X})) = \widetilde{pre}[\tau](\{s \mid \mathcal{M}_\tau\downarrow_s \subseteq \mathcal{X}\}) = \widetilde{pre}[\tau](\{\bullet\}) = \{s \mid \forall s' : t(s, s') \Rightarrow s' = \bullet\} = \emptyset$ since $t(s, \bullet)$ implies $s = \bullet$ and $t(\bullet, \circ)$ holds. ∎

11.3 Commutation of the reverse operator

For the reverse operator, we have:

$$\alpha^\vee_{\mathcal{M}_\tau}(\curvearrowleft\llbracket\mathcal{X}\rrbracket) = \alpha^\vee_{\mathcal{M}_{\tau^{-1}}}(\mathcal{X}) \qquad \textbf{(57)}$$

PROOF

$\alpha^\vee_{\mathcal{M}_\tau}(\curvearrowleft\llbracket\mathcal{X}\rrbracket)$

$= \{s \mid \mathcal{M}_\tau\downarrow_s \subseteq (\curvearrowleft\llbracket\mathcal{X}\rrbracket)\}$ {by Definition (46) of $\alpha^\vee_{\mathcal{M}_\tau}$}

$=$ {by Definition (3) of $\curvearrowleft\llbracket\bullet\rrbracket$ and that of $\subseteq$}

$\{s \mid \curvearrowleft\llbracket\mathcal{M}_\tau\downarrow_s\rrbracket \subseteq (\curvearrowleft\llbracket\curvearrowleft\llbracket\mathcal{X}\rrbracket\rrbracket)\}$

$=$ {by Definition (3) of $\curvearrowleft\llbracket\bullet\rrbracket$ so that $\curvearrowleft\llbracket\curvearrowleft\llbracket M\rrbracket\rrbracket = M$}

$\{s \mid \curvearrowleft\llbracket\mathcal{M}_\tau\downarrow_s\rrbracket \subseteq \mathcal{X})\}$

$=$ {by Definition (11) of $\mathcal{M}_\tau$, (3) of $\curvearrowleft\llbracket\bullet\rrbracket$, (4) of $\bullet\downarrow_\bullet$ and Lemma (17)}

$\{s \mid \mathcal{M}_{\tau^{-1}}\downarrow_s \subseteq \mathcal{X})\}$

$= \alpha^\vee_{\mathcal{M}_{\tau^{-1}}}(\mathcal{X})$ {by Definition (46) of $\alpha^\vee_{\mathcal{M}_\tau}$} ∎

11.4 Semi-commutation for the successor operator

In order to illustrate the time-symmetric reasoning, let us consider:

$\alpha^\vee_{\mathcal{M}_\tau}(\ominus\llbracket\mathcal{X}\rrbracket)$

$= \alpha^\vee_{\mathcal{M}_\tau}(\curvearrowleft\llbracket\oplus\llbracket\curvearrowleft\llbracket\mathcal{X}\rrbracket\rrbracket\rrbracket)$ {by Definition (7) of $\ominus\llbracket\bullet\rrbracket$}

$= \alpha^\vee_{\mathcal{M}_{\tau^{-1}}}(\oplus\llbracket\curvearrowleft\llbracket\mathcal{X}\rrbracket\rrbracket)$ {by (57)}

$\supseteq / \subseteq_{FD(\curvearrowleft\llbracket\mathcal{X}\rrbracket)}$ i.e. $\supseteq / \subseteq_{BD(\mathcal{X})}$ $\widetilde{pre}[\tau^{-1}](\alpha^\vee_{\mathcal{M}_{\tau^{-1}}}(\curvearrowleft\llbracket\mathcal{X}\rrbracket))$ {by (51)}

$= \widetilde{pre}[\tau^{-1}](\alpha^\vee_{\mathcal{M}_{(\tau^{-1})^{-1}}}(\mathcal{X}))$ {by (57)}

$= \widetilde{post}[\tau](\alpha^\vee_{\mathcal{M}_\tau}(\mathcal{X}))$ {by $(\tau^{-1})^{-1} = \tau$, Defs. (43) and (44)} ∎

11.5 Commutation of the conjunction operator

By (48), $\alpha^\vee_{\mathcal{M}_\tau}$ is a complete $\supseteq$-join morphism that is a complete $\cap$-morphism $\alpha^\vee_{\mathcal{M}_\tau}\left(\bigcap_{i\in\Delta}\mathcal{X}_i\right) = \bigcap_{i\in\Delta}\alpha^\vee_{\mathcal{M}_\tau}(\mathcal{X}_i)$. It follows that the commutation Hypothesis (32) holds for $\cap\llbracket\bullet, \bullet\rrbracket$.

11.6 (Semi-)commutation of the disjunction operator

By (48), $\alpha^\vee_{\mathcal{M}_\tau}$ is $\supseteq$-monotone hence $\subseteq$-monotone so that $\alpha^\vee_{\mathcal{M}_\tau}(\mathcal{X}_1) \cup \alpha^\vee_{\mathcal{M}_\tau}(\mathcal{X}_2) \subseteq \alpha^\vee_{\mathcal{M}_\tau}(\mathcal{X}_1 \cup \mathcal{X}_2)$. However, in general, equality does not hold, even for forward temporal models:

$$\alpha^\vee_{\mathcal{M}_\tau}(\mathcal{X}_1) \cup \alpha^\vee_{\mathcal{M}_\tau}(\mathcal{X}_2) \subsetneq \alpha^\vee_{\mathcal{M}_\tau}(\mathcal{X}_1 \cup \mathcal{X}_2) \qquad \textbf{(58)}$$

PROOF Consider the counter-example $\mathbb{S} \triangleq \{\circ, \bullet\}$, $\tau \triangleq \{\langle\bullet, \bullet\rangle, \langle\bullet, \circ\rangle, \langle\circ, \circ\rangle\}$ so that $\mathcal{M}_\tau = \{\langle i, \lambda\ell\cdot\bullet\rangle, \langle j, \lambda\ell\cdot\circ\rangle, \langle k, \lambda\ell\cdot(\ell < m\ ?\ \bullet\ ¿\ \circ)\rangle\}$. We let $\mathcal{X}_1 \triangleq \Diamond\llbracket\boxplus\llbracket\{\circ\}\rrbracket\rrbracket$ and $\mathcal{X}_2 \triangleq \boxplus\llbracket\{\bullet\}\rrbracket$ which are both forward closed temporal models. We have $\alpha^\vee_{\mathcal{M}_\tau}(\mathcal{X}_1) = \{\circ\}$ since in a $\bullet$-state, one may always stay in a $\bullet$-state whence never satisfy $\boxplus\llbracket\circ\rrbracket$. $\alpha^\vee_{\mathcal{M}_\tau}(\mathcal{X}_2) = \emptyset$ since the present state must be $\bullet$ which does not prevent some future state to be $\circ$. However, we have $\alpha^\vee_{\mathcal{M}_\tau}(\mathcal{X}_1 \cup \mathcal{X}_2) = \{\circ, \bullet\}$ since in a $\circ$-state, the future states will all be $\circ$ so that $\boxplus\llbracket\{\circ\}\rrbracket$ hence $\Diamond\llbracket\boxplus\llbracket\{\circ\}\rrbracket\rrbracket$ holds. Otherwise in a $\bullet$-state either all future states will be $\bullet$ in which case $\boxplus\llbracket\{\bullet\}\rrbracket$ holds or a $\langle\bullet, \circ\rangle$-transition will be fired and later states will all be $\circ$ in which case $\Diamond\llbracket\boxplus\llbracket\{\circ\}\rrbracket\rrbracket$ holds. ∎ Nevertheless, when one of $\mathcal{X}_1$ or $\mathcal{X}_2$ is state-closed then the commutation Hypothesis (32) holds for $\cup\llbracket\bullet, \bullet\rrbracket$:

$$ST(\mathcal{X}_1) \vee ST(\mathcal{X}_2) \Rightarrow \left((\alpha^\vee_{\mathcal{M}_\tau}(\mathcal{X}_1) \cup \alpha^\vee_{\mathcal{M}_\tau}(\mathcal{X}_2) = \alpha^\vee_{\mathcal{M}_\tau}(\mathcal{X}_1 \cup \mathcal{X}_2)\right) \qquad \textbf{(59)}$$

PROOF By commutativity, we only have to consider the case when $ST(\mathcal{X}_1)$. By inclusion (58), we only have to prove that $\alpha^\vee_{\mathcal{M}_\tau}(\mathcal{X}_1) \cup \alpha^\vee_{\mathcal{M}_\tau}(\mathcal{X}_2) \supseteq \alpha^\vee_{\mathcal{M}_\tau}(\mathcal{X}_1 \cup \mathcal{X}_2)$, which we do by reductio ad absurdum.

If inclusion (58) is strict then there exists a state s such that $s \in \alpha^\forall_{\mathcal{M}_\tau}(\mathcal{X}_1 \cup \mathcal{X}_2) \cup \neg\alpha^\forall_{\mathcal{M}_\tau}(\mathcal{X}_1) \cup \neg\alpha^\forall_{\mathcal{M}_\tau}(\mathcal{X}_2)$. By Definition (46) of $\alpha^\forall_{\mathcal{M}_\tau}$, $s \in \alpha^\forall_{\mathcal{M}_\tau}(\mathcal{X}_1 \cup \mathcal{X}_2)$ implies that $\forall \langle i, \sigma\rangle \in \mathcal{M}_\tau : \sigma_i = s \Rightarrow \langle i, \sigma\rangle \in (\mathcal{X}_1 \cup \mathcal{X}_2)$. The implication cannot vacuously hold since there exists some $\langle i, \sigma\rangle \in \mathcal{M}_\tau$ satisfying $\sigma_i = s$ since otherwise $\mathcal{M}_\tau\!\downarrow_s = \emptyset$ in contradiction with Lemma (12). Hence for any such $\langle i, \sigma\rangle$, either $\langle i, \sigma\rangle \in \mathcal{X}_1$ or $\langle i, \sigma\rangle \in \mathcal{X}_2$. $\mathcal{X}_1$ is state-closed so if $\langle i, \sigma\rangle \in \mathcal{X}_1$ then any other $\langle j, \sigma'\rangle$ with $\sigma_j = \sigma_i = s$ also belongs to $\mathcal{X}_1$ proving that $\mathcal{M}_\tau\!\downarrow_s \subseteq \mathcal{X}_1$ in contradiction with $s \notin \alpha^\forall_{\mathcal{M}_\tau}(\mathcal{X}_1)$. We conclude that the only possibility is $\langle i, \sigma\rangle \in \mathcal{X}_2$, whence $\forall \langle i, \sigma\rangle \in \mathcal{M}_\tau : \sigma_i = s \Rightarrow \langle i, \sigma\rangle \in \mathcal{X}_2$ which now is in contradiction with $s \notin \alpha^\forall_{\mathcal{M}_\tau}(\mathcal{X}_2)$. ∎

11.7 Abstraction of the model complement

For the negation operator, we observe that:

$$\alpha^\forall_{\mathcal{M}_\tau}(\neg\{\!|\mathcal{X}|\!\}) = \neg\alpha^\exists_{\mathcal{M}_\tau}(\mathcal{X})$$

PROOF

$$\begin{aligned}
& \alpha^\forall_{\mathcal{M}_\tau}(\neg\{\!|\mathcal{X}|\!\}) \\
= {} & \{s \mid \mathcal{M}_\tau\!\downarrow_s \subseteq (\neg\{\!|\mathcal{X}|\!\})\} && \wr\text{by Definition (46) of } \alpha^\forall_{\mathcal{M}_\tau}\wr \\
= {} & \{s \mid \mathcal{M}_\tau\!\downarrow_s \subseteq (\neg\mathcal{X})\} && \wr\text{by Definition (6) of } \neg\{\!|\bullet|\!\}\wr \\
= {} & \neg\{s \mid \neg(\mathcal{M}_\tau\!\downarrow_s \subseteq (\neg\mathcal{X}))\} && \wr\text{by Definition of } \neg\wr \\
= {} & \neg\{s \mid \mathcal{M}_\tau\!\downarrow_s \cap \mathcal{X}) \neq \emptyset\} && \wr\text{by Definition of } \cap\wr \\
= {} & \neg\alpha^\exists_{\mathcal{M}_\tau}(\mathcal{X}) && \wr\text{by Definition (49) of } \alpha^\exists_{\mathcal{M}_\tau}\wr \ ∎
\end{aligned}$$

11.8 Commutation of the universal state closure operator

As seen in Section 5, the universal state closure operator is $\forall\varphi_1 \triangleq \forall\boxplus(\pi_\tau) : \varphi_1$ so that, when considering the semantics of these formulae in Definition (13), we have

$$\begin{aligned}
& \forall\{\!|\mathcal{X}|\!\} \\
\triangleq {} & \forall\{\!|\mathcal{M}_\tau, \mathcal{X}|\!\} && \wr\text{by Definitions (9) and (11)}\wr \\
= {} & \{\langle i, \sigma\rangle \in \mathcal{M}_\tau \mid \mathcal{M}_\tau\!\downarrow_{\sigma_i} \subseteq \mathcal{X}\} && \wr\text{by Definition (5)}\wr \\
= {} & \{\langle i, \sigma\rangle \in \mathcal{M}_\tau \mid \sigma_i \in \{s \mid \mathcal{M}_\tau\!\downarrow_s \subseteq \mathcal{X}\}\} \\
= {} & \{\langle i, \sigma\rangle \in \mathcal{M}_\tau \mid \sigma_i \in \alpha^\forall_{\mathcal{M}_\tau}(\mathcal{X})\} && \wr\text{by definition (46)}\wr \\
= {} & \gamma^\forall_{\mathcal{M}_\tau}(\alpha^\forall_{\mathcal{M}_\tau}(\mathcal{X})) && \wr\text{by definition (47)}\wr
\end{aligned}$$

It follows that $\forall$ is the upper closure operator $\gamma^\forall_{\mathcal{M}_\tau} \circ \alpha^\forall_{\mathcal{M}_\tau}$ whence our "universal state closure" terminology. The commutation condition of Hypothesis (32) follows immediately since:

$$\begin{aligned}
& \alpha^\forall_{\mathcal{M}_\tau} \circ \forall\{\!|\mathcal{X}|\!\} \\
= {} & \alpha^\forall_{\mathcal{M}_\tau} \circ \gamma^\forall_{\mathcal{M}_\tau} \circ \alpha^\forall_{\mathcal{M}_\tau} \\
= {} & \alpha^\forall_{\mathcal{M}_\tau} && \wr\text{by Galois connection (48)}\wr \\
= {} & 1 \circ \alpha^\forall_{\mathcal{M}_\tau} && \wr\text{by Definition of the identity map } 1\wr
\end{aligned}$$

12. Incompleteness of the transition checking abstraction of temporal calculi/logics for finite systems

It follows from the previous section, that the universal (and dually existential) checking abstractions do not satisfy the sufficient conditions of Theorem (40) for completeness. Indeed these abstractions are not complete, even for finite transition systems $\langle \mathbb{S}, \tau\rangle$ as shown by the following trivial counter-example. This means that in general, using the reversible $\widehat{\mu}^\star$-calculus with trace-based and set-based semantics is not equivalent.

Counter example (60) Let us consider the formula $\ominus\oplus\boxminus\sigma_{\{\bullet\}}$ for the transition system $\mathbb{S} = \{\circ, \bullet\}$ and $\tau = \{\langle\bullet, \bullet\rangle, \langle\bullet, \circ\rangle, \langle\circ, \circ\rangle\}$. On one hand, by Definition (13) and (8), the concrete semantics is $[\![\ominus\oplus\boxminus\sigma_{\{\bullet\}}]\!]\emptyset = \ominus\{\!|\oplus\{\!|\mathrm{gfp}^{\subseteq}\lambda X \cdot \cap\{\!|\sigma_{\{\bullet\}}, \ominus\{\!|X|\!\}|\!\}|\!\}|\!\} = \{\langle i, \sigma\rangle \mid \forall j \leq i : \sigma_j = \bullet\}$ so that its universal state abstraction is $\alpha^\forall_{\mathcal{M}_\tau}[\![\ominus\oplus\boxminus\sigma_{\{\bullet\}}]\!]\emptyset = \alpha^\forall_{\mathcal{M}_\tau}\{\langle i, \sigma\rangle \mid \forall j \leq i : \sigma_i = \bullet\} = \{\bullet\}$. On the other hand, the abstract semantics is $[\![\ominus\oplus\boxminus\sigma_{\{\bullet\}}]\!]^\forall = \widetilde{pre}[\tau](\widetilde{post}[\tau](\mathrm{gfp}^{\subseteq}\lambda X \cdot \cap\{\!|\sigma_{\{\bullet\}}, \widetilde{post}[\tau](X)|\!\})) = \widetilde{pre}[\tau](\widetilde{post}[\tau](\{\bullet\})) = \widetilde{pre}[\tau](\{\bullet\}) = \emptyset$. So $\alpha^\forall_{\mathcal{M}_\tau}([\![\ominus\oplus\boxminus\sigma_{\{\bullet\}}]\!]\emptyset) \supsetneq [\![\ominus\oplus\boxminus\sigma_{\{\bullet\}}]\!]^\forall$.

13. Completeness of the transition checking abstraction for the μ_+-calculus and CTL

Incompleteness results from the predecessor/successor operators when mixing forward and backward modalities (56) as well as from general disjunction (58). Completeness is obtained by considering forward modalities only as well as the restriction of disjunction to at least one state formulae (59). We get the $\mu^\forall_+$-calculus:

$$\begin{aligned}
\psi &::= \sigma_S \mid \psi_1 \vee \psi_2 \mid \psi_1 \wedge \psi_2 \mid \neg\psi_1 \mid \forall\varphi && \text{state formulae} \\
\varphi &::= \psi \mid \pi_t \mid \oplus\varphi_1 \mid \varphi_1 \wedge \varphi_2 \mid && \text{path formulae} \\
&\qquad \psi_1 \vee \varphi_2 \mid \varphi_1 \vee \psi_2 \mid X \mid \mu X \cdot \varphi_1 \mid \nu X \cdot \varphi_1
\end{aligned}$$

such that $[\![\varphi]\!]^\forall = \vec{\alpha}^\forall_{\mathcal{M}_\tau}([\![\varphi]\!])$. This covers $\forall$CTL$_+$ whence $\forall$CTL$_-$ by reversal. Dual results hold for the existential abstraction which yields a completeness result for $\exists$CTL$_+$ and $\exists$CTL$_-$. Observe that contrary to classical comparable results [4, 5, 16], there is no obligation to approximate the transition system by an abstract transition system, as is the case in standard abstract model checking.

14. Abstract model checking

Abstract model checking [4, 5] uses a set $\bar{\mathbb{S}}$ of abstract states (so that the corresponding set of abstract models is $\bar{\mathbb{M}} \triangleq \wp(\mathbb{Z} \times (\mathbb{Z} \mapsto \bar{\mathbb{S}})))$ as well as state property and model abstractions:

$$\langle \wp(\mathbb{S}), \supseteq\rangle \underset{\bar{\alpha}^\forall_{\mathrm{s}}}{\overset{\bar{\gamma}^\forall_{\mathrm{s}}}{\leftrightarrows}} \langle \wp(\bar{\mathbb{S}}), \subseteq\rangle \qquad \bar{\alpha}^\forall_{\mathrm{s}} \in \wp(\mathbb{S}) \xmapsto{\perp,\mathrm{con}} \wp(\bar{\mathbb{S}}) \qquad \textbf{(61)}$$

$$\langle \mathbb{M}, \subseteq\rangle \underset{\bar{\alpha}^\forall_{\mathrm{m}}}{\overset{\bar{\gamma}^\forall_{\mathrm{m}}}{\leftrightarrows}} \langle \bar{\mathbb{M}}, \subseteq\rangle \qquad \bar{\gamma}^\forall_{\mathrm{m}} \in \mathbb{M} \xmapsto{\perp,\mathrm{con}} \bar{\mathbb{M}} \qquad \textbf{(62)}$$

The corresponding universal checking abstraction:

$$\bar{\alpha}^\forall_M \triangleq \bar{\alpha}^\forall_{\mathrm{s}} \circ \alpha^\forall_M \circ \bar{\gamma}^\forall_{\mathrm{m}} \qquad \bar{\gamma}^\forall_M \triangleq \bar{\gamma}^\forall_{\mathrm{m}} \circ \alpha^\forall_M \circ \bar{\gamma}^\forall_{\mathrm{s}} \qquad \textbf{(63)}$$

satisfies (48).

In the literature on abstract model checking [4, 5], a single type of self-dual abstraction is considered, given by $@ \in \mathbb{S} \mapsto \bar{\mathbb{S}}$, so that:

$$\begin{aligned}
\bar{\alpha}^\forall_{\mathrm{s}}(S) &\triangleq \{@(s) \mid s \in S\} \qquad \bar{\gamma}^\forall_{\mathrm{s}}(\bar{S}) \triangleq \{s \mid @(s) \in \bar{S}\} \\
\bar{\alpha}^\forall_{\mathrm{m}}(M) &\triangleq \{\langle i, \lambda j \cdot @(\sigma_j)\rangle \mid \langle i, \sigma\rangle \in M\} \\
\bar{\gamma}^\forall_{\mathrm{m}}(\bar{M}) &\triangleq \{\langle i, \sigma\rangle \mid \langle i, \lambda j \cdot @(\sigma_j)\rangle \in \bar{M}\}
\end{aligned}$$

which respectively satisfy (61) and (62). By defining an abstract transition system:

$$\bar{\tau} \triangleq \{\langle \bar{s}_1, \bar{s}_2\rangle \mid \exists s_1, s_2 : @(s_1) = \bar{s}_1 \wedge @(s_{21}) = \bar{s}_2 \wedge \tau(s_1, s_2)\} \qquad \textbf{(64)}$$

we have $\mathcal{M}_{\bar{\tau}} \supseteq \bar{\alpha}^{\forall}_{\mathrm{m}}(\mathcal{M}_{\tau})$ whence $\bar{\alpha}^{\forall}_{\mathrm{m}}(\mathcal{M}_{\tau\downarrow s}) \subseteq (\bar{\alpha}^{\forall}_{\mathrm{m}}(\mathcal{M}_{\tau})_{\downarrow @(s)} \subseteq \mathcal{M}_{\bar{\tau}\downarrow @(s)}$ so that:

$$\begin{aligned}
&\bar{\alpha}^{\forall}_{\mathcal{M}_\tau}(\phi) \\
=\ & \bar{\alpha}^{\forall}_{\mathrm{s}} \circ \alpha^{\forall}_{\mathcal{M}_\tau} \circ \bar{\gamma}^{\forall}_{\mathrm{m}}(\phi) && \wr\text{by (63)}\int \\
=\ & \bar{\alpha}^{\forall}_{\mathrm{s}}(\{s \mid \mathcal{M}_{\tau\downarrow s} \subseteq \bar{\gamma}^{\forall}_{\mathrm{m}}(\phi)\}) && \wr\text{by (46)}\int \\
=\ & \{@(s) \mid \bar{\alpha}^{\forall}_{\mathrm{m}}(\mathcal{M}_{\tau\downarrow s}) \subseteq \phi\} && \wr\text{by (61) and (62)}\int \\
=\ & \{\bar{s} \mid \mathcal{M}_{\bar{\tau}\downarrow s} \subseteq \phi\} && \wr\text{since } \bar{\alpha}^{\forall}_{\mathrm{m}}(\mathcal{M}_{\tau\downarrow s}) \subseteq \mathcal{M}_{\bar{\tau}\downarrow @(s)}\int \quad \blacksquare
\end{aligned}$$

which is of the required form (46) for the results on checking abstractions to be directly applicable. Dual definitions, hypotheses and results hold for the existential abstraction $\bar{\alpha}^{\exists}_{M}$ satisfying (50). However this is a very restricted form of abstraction, mainly useful to reuse existing model-checkers. Moreover in practice $\bar{\mathbb{S}}$ must be finite, which in view of [10], requires the abstraction to be redesigned for each particular concrete transition system, which on one hand might not be of this restricted form and on the other hand is unthinkable in the context of program analysis.

15. Data-flow analysis is a boolean abstract interpretation

Data flow analysis was shown, at least for the available expressions example, to be a boolean abstract interpretation in [9, example 7.2.0.6.3]. In [9], the semantics of programs is a (prefix closed) set of (finite) traces generated by a transition system. The data flow property specification is also a set of traces specified equationally i.e. inductively along one path. The abstraction is the composition of a static partitioning with a checking abstraction including the existential or universal merging of path properties. The contribution of [23] is essentially in the use of a branching time temporal logic for the data flow property specification so that model-checkers boolean equations solvers can be reused for dataflow analysis. [22] remarks that the abstract flowchart with respect to which the data flow property is specified by [23] is itself an abstract interpretation of the program semantics. This essentially consists in applying the static partitioning abstraction of [6]. A problem with this "data-flow analysis as model checking of abstract interpretations" approach is that the abstract interpretation of the program semantics into an abstract flowchart and the model-checking specification of the abstract flowchart are separate processes. Their composition may be not trivial as shown by live-variable analysis which is erroneously claimed to be unsound in [22]. By understanding both processes as abstract interpretations, their combination becomes a well-understood composition of abstract interpretations.

15.1 The data-flow analysis abstractions

Assume that programs have a finite set of labels $\ell \in \mathcal{L}$ such that the set $\mathbb{S}$ of states can be partitioned into $\{\mathbb{S}_\ell \mid \ell \in \mathcal{L}\}$ (i.e. $\mathbb{S} = \bigcup_{\ell\in\mathcal{L}} \mathbb{S}_\ell$ and $\forall \ell, \ell' \in \mathcal{L} : (\ell \neq \ell') \Rightarrow (\mathbb{S}_\ell \cap \mathbb{S}_{\ell'} = \emptyset)$). The boolean version of the *static partitioning abstraction* of [6] is:

$$\begin{aligned}
&\alpha^{\forall}_{\mathcal{L}}(S) \triangleq \prod_{\ell\in\mathcal{L}} \mathbb{S}_\ell \subseteq S && \gamma^{\forall}_{\mathcal{L}}(B) \triangleq \bigcup\{\mathbb{S}_\ell \mid B(\ell)\} && \text{universal} \\
&\alpha^{\exists}_{\mathcal{L}}(S) \triangleq \prod_{\ell\in\mathcal{L}} (\mathbb{S}_\ell \cap S) \neq \emptyset && \gamma^{\exists}_{\mathcal{L}}(B) \triangleq \bigcap\{\mathbb{S}_\ell \mid B(\ell)\} && \text{existential}
\end{aligned}$$

so that:

$$\langle \wp(\mathbb{S}), \supseteq\rangle \underset{\alpha^{\forall}_{\mathcal{L}}}{\overset{\gamma^{\forall}_{\mathcal{L}}}{\leftrightarrows}} \langle \prod_{\ell\in\mathcal{L}} \mathbb{B}, \dot{\Leftarrow}\rangle \qquad \langle \wp(\mathbb{S}), \subseteq\rangle \underset{\alpha^{\exists}_{\mathcal{L}}}{\overset{\gamma^{\exists}_{\mathcal{L}}}{\leftrightarrows}} \langle \prod_{\ell\in\mathcal{L}} \mathbb{B}, \dot{\Rightarrow}\rangle$$

which can be composed with the checking abstractions of Section 10:

$$\begin{aligned}
\dot{\alpha}^{\forall}_{\mathcal{L},\tau} &\triangleq \alpha^{\forall}_{\mathcal{L}} \circ \alpha^{\forall}_{\mathcal{M}_\tau} & \dot{\alpha}^{\exists}_{\mathcal{L},\tau} &\triangleq \alpha^{\exists}_{\mathcal{L}} \circ \alpha^{\exists}_{\mathcal{M}_\tau} \\
&= \prod_{\ell\in\mathcal{L}} (\mathcal{M}_\tau \cap \sigma\{\!\!\{\mathbb{S}_\ell\}\!\!\}) \subseteq \phi & &= \prod_{\ell\in\mathcal{L}} (\mathcal{M}_\tau \cap \sigma\{\!\!\{\mathbb{S}_\ell\}\!\!\} \cap \phi) \neq \emptyset
\end{aligned}$$

The boolean dataflow equations can then be designed by calculus starting from the temporal specification as given e.g. in [22].

15.2 On live-variable data flow analysis

Assume that $\langle \mathbb{S}, \tau\rangle$ is the small-step operational semantics of the program where states have the form $s = \langle s_\ell, s_\Gamma\rangle \in \mathbb{S} \triangleq \mathcal{L} \times (\mathbb{X} \mapsto \mathbb{V})$ where s_ℓ is the label of state s and the environment s_Γ of state s assigns values $s_\Gamma(\mathrm{x})$ to program variables x. Let $\mathsf{mod}(\mathrm{x})$ be a specification of the program variables x which are potentially modified by a step τ and $\mathsf{used}(\mathrm{x})$ be a characterization of the transitions τ definitely using the value of x:

$$\begin{aligned}
\mathsf{mod}(\mathrm{x}) &\triangleq \{\langle s, s'\rangle \mid \tau(s, s') \wedge s'_\Gamma(\mathrm{x}) \neq s_\Gamma(\mathrm{x})\} \\
\mathsf{used}(\mathrm{x}) &\triangleq \{\langle s, s'\rangle \mid \tau(s, s') \wedge \exists v \in \mathbb{V} : \neg\tau(\langle s_\ell, s_\Gamma[\mathrm{x} := v]\rangle, s')\}
\end{aligned}$$

"In *live-variable* analysis we wish to know for variable x and point p whether the value of x at p could be used along some path in the flow graph starting at p. If so, we say x is *live* at p; otherwise x is *dead* at p" [1, p. 631]. From this informal specification, we derive that variable x is live at the origin of a computation if and only if it will not be modified until it is used:

$$\mathsf{isLive}(\mathrm{x}) \triangleq (\neg\,\pi\{\!\!\{\mathsf{mod}(\mathrm{x})\}\!\!\})\ \mathbf{U}\ \pi\{\!\!\{\mathsf{used}(\mathrm{x})\}\!\!\}$$

Now, variable x is live at location ℓ if and only if it is live on <u>some</u> computation path starting from that location ℓ. It is dead if and only if it is not live:

$$\begin{aligned}
\mathit{Live}(\mathrm{x}) &\triangleq \dot{\alpha}^{\exists}_{\mathcal{L},\tau}([\![\mathsf{isLive}(\mathrm{x})]\!]\emptyset) \\
\mathit{Dead}(\mathrm{x}) &\triangleq \dot{\neg}\,\mathit{Live}(\mathrm{x}) = \dot{\alpha}^{\forall}_{\mathcal{L},\tau}(\neg[\![\mathsf{isLive}(\mathrm{x})]\!]\emptyset)
\end{aligned} \tag{65}$$

The classical dataflow equations derives from this specification using Theorem (40). We define:

$$\begin{aligned}
\mathsf{succ}(l) &\supseteq \{\ell' \mid \exists s \in \mathbb{S}_\ell : \exists s' \in \mathbb{S}_{\ell'} : \tau(s, s')\} \\
\mathsf{used}^{\sharp}_{\mathrm{x}}(l, l') &\Leftarrow \exists s \in \mathbb{S}_\ell : \exists s' \in \mathbb{S}_{\ell'} : \langle s, s'\rangle \in \mathsf{used}(\mathrm{x}) \\
\neg\mathsf{mod}^{\sharp}_{\mathrm{x}}(l, l') &\Leftarrow \exists s \in \mathbb{S}_\ell : \exists s' \in \mathbb{S}_{\ell'} : \langle s, s'\rangle \notin \mathsf{mod}(\mathrm{x})
\end{aligned}$$

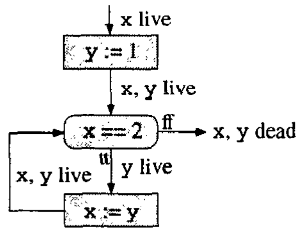
so that:

$$\begin{aligned}
&\mathit{Live}(\mathrm{x}) \dot{\Rightarrow} \mathit{Live}^{\sharp}(\mathrm{x}) \triangleq \\
&\mathsf{lfp}^{\dot{\Rightarrow}}\, \lambda X \cdot \lambda l \cdot (\exists l' : \mathsf{used}^{\sharp}_{\mathrm{x}}(l, l')) \vee \bigvee_{l' \in \mathsf{succ}(l)} \neg\mathsf{mod}^{\sharp}_{\mathrm{x}}(l, l') \wedge X(l')
\end{aligned} \tag{66}$$

Notice the implication $\dot{\Rightarrow}$, not $\dot{\Leftarrow}$, so liveness is about "might be" live variables. By defining $\mathit{Dead}^{\sharp}(\mathrm{x}) \triangleq \dot{\neg}\,\mathit{Live}^{\sharp}(\mathrm{x})$ we have $\mathit{Dead}^{\sharp}(\mathrm{x}) \dot{\Rightarrow} \mathit{Dead}(\mathrm{x})$ proving that the detection of "must be" dead variables is always correct.

15.3 On the soundness of live-variable analysis

The example given by [22, Sec. 7, Fig. 5] to show that live variable analysis is unsound is reproduced on next page. The argument is that y is declared live by the classical data flow analysis equations (66) although y is not live at the program entry on the concrete execution path starting with x = 2. This is harmless since "data-flow practitioners are well aware of the above problem, and disaster does not arise in practice, because live variables analysis is used 'dually' — it is used to detect *dead variables*". Observe that the example is correct with respect to the specification (65) and (66). The unsoundness argument in [22] follows from a (mis-)specification of live variable analysis as a universal abstraction $Live(\mathsf{x}) \not\overset{\Delta}{=} \dot{\alpha}^{\forall}_{\mathcal{L},\tau}([\![\mathsf{isLive}(\mathsf{x})]\!]\emptyset)$. Then definition of dead variables as $Dead^{\sharp}(\mathsf{x}) \overset{\Delta}{=} \neg\, Live^{\sharp}(\mathsf{x})$ in [22] is existential whence also invalid. However the fixpoint characterization given by [22] for dead variables, which can be easily derived from (66) by Park dual fixpoint theorem, is correct. Hence the correct classical data flow equations (both (66) for live variables and their dual for dead variables) cannot be derived from the incorrect modal specification (indeed the universal abstraction for live variables would lead to a greatest fixpoint). This incoherence in [22] shows the importance of formally deriving the model checking equations by abstract interpretation of the temporal logic specification.



Example of [22, Sec. 7, Fig. 5]

16. Conclusion

Model checking of finite systems is understood as a complete formal verification method as opposed to static analysis/testing of infinite state programs which is fundamentally incomplete. In practice this does not make a significative difference because the systems which are verified are relatively small when compared e.g., in hardware, with the size of a microprocessor. So the success stories in model checking are most often relative to bugs which have been found after exploration of part of the state space. When considering infinite systems, and except for very particular cases, incompleteness is to be taken into account. As is the case in program analysis by abstract interpretation, this does not prevent infinite systems to be verified, but partially only, that is by restricting the abstract properties that can be checked. Our understanding of model-checking as an abstract interpretation leads to three possible research directions different from the present-day manual design of the abstraction [4, 5, 16]. First, one can look for generic temporal abstract domains going well beyond the abstract transition systems of (64) Section 14, as in [21]. Second, the optimal abstract domain for a particular property of a particular transition system can be formally specified [15] and it might be the case that its computation could, at least partially, be automated. Third, [10] shows that the general abstract interpretation approaches applicable to families of systems must abandon the consideration of finite abstract domains (e.g. [5, 12]) and use widening/narrowing techniques [8] which has not yet been much explored, except in the classical application of polyhedral abstract interpretation [11] to the verification of safety properties [13, 17].

References

[1] AHO, A., SETHI, R., AND ULLMAN, J. *Compilers. Principles, Technique and Tools*. Addison-Wesley, 1986.

[2] BEN-ARI, M., MANNA, Z., AND PNUELI, A. The temporal logic of branching time. *Acta Informat. 20* (1983), 207–226.

[3] CLARKE, E., EMERSON, E., AND SISTLA, A. Automatic verification of finite-state concurrent systems using temporal logic specifications. *TOPLAS 8*, 2 (Jan. 1986), 244–263.

[4] CLARKE, E., GRUMBERG, O., AND LONG, D. Model checking and abstraction. *TOPLAS 16*, 5 (Sep. 1994), 1512–1542.

[5] CLEAVELAND, R., IYER, P., AND YANKELEVITCH, D. Optimality in abstractions of model checking. *Proc. 2nd Int. Symp. SAS '95*, LNCS 983. Springer-Verlag, 1995, 51–63.

[6] COUSOT, P. Semantic foundations of program analysis. In *Program Flow Analysis: Theory and Applications*, S. Muchnick & N. Jones, Eds. Prentice-Hall, 1981, 303–342.

[7] COUSOT, P. Types as abstract interpretations, invited paper. *24th POPL* (Paris, 1997), 316–331.

[8] COUSOT, P., AND COUSOT, R. Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints. *4th POPL* (Los Angeles, 1977), 238–252.

[9] COUSOT, P., AND COUSOT, R. Systematic design of program analysis frameworks. *6th POPL* (San Antonio, 1979), 269–282.

[10] COUSOT, P., AND COUSOT, R. Comparing the Galois connection and widening/narrowing approaches to abstract interpretation, invited paper. *Proc. Int. Work. PLILP '92*, LNCS 631. Springer-Verlag, 1992, 269–295.

[11] COUSOT, P., AND HALBWACHS, N. Automatic discovery of linear restraints among variables of a program. *5th POPL* (Tucson, 1978), 84–97.

[12] DAMS, D., GRUMBERG, O., AND GERTH, R. Abstract interpretation of reactive systems. *TOPLAS 19*, 2 (1997), 253–291.

[13] DILL, D., AND WONG-TOI, H. Verification of real-time systems by successive over and under approximation. *Proc. 7th Int. Conf. CAV '95*, LNCS 939. Springer-Verlag, 1995, 409–422.

[14] EMERSON, E., AND HALPERN, J. "Sometimes" and "Not Never" revisited: On branching time versus linear time. *TOPLAS 33* (1986), 151–178.

[15] GIACOBAZZI, R., RANZATO, F., AND SCOZZARI, F. Complete abstract interpretations made constructive. *Proc. 23rd Int. Symp. MFCS '98*, LNCS 1450. Springer-Verlag, 1998, 366–377.

[16] GRAF, S., AND LOISEAUX, C. A tool for symbolic program verification and abstraction. *Proc. 5th Int. Conf. CAV '93*, LNCS 697. Springer-Verlag, 1993, 71–84.

[17] HALBWACHS, N., PROY, Y., AND ROUMANOFF, P. Verification of real-time systems using linear relation analysis. *Formal Methods in System Design 11*, 2 (Aug. 1997), 157–185.

[18] KOZEN, D. Results on the propositional μ-calculus. *Theoret. Comput. Sci. 27* (1983), 333–354.

[19] LONG, D., BROWNE, A., CLARKE, E., JHA, S., AND MARRERO, W. An improved algorithm for the evaluation of fixpoint expressions. *Theoret. Comput. Sci.* 178(1-2):237-255 (1997).

[20] MANNA, Z., AND PNUELI, A. *The Temporal Logic of Reactive and Concurrent Systems, Specification.* Springer-Verlag, 1992.

[21] MAUBORGNE, L. Binary decision graphs. *Proc. 6th Int. Symp. SAS '99*, LNCS 1694. Springer-Verlag, 1999, 101–116.

[22] SCHMIDT, D. Data-flow analysis is model checking of abstract interpretations. *25th POPL* (San Diego, 1998), 38–48.

[23] STEFFEN, B. Generating data flow analysis algorithms from modal specifications. *Sci. Comput. Programming 21* (1993), 115–139.