Partial model checking via abstract interpretation

N. De Francesco, G. Lettieri*, L. Martini, G. Vaglini

Università di Pisa, Dipartimento di Ingegneria dell'Informazione, sez. Informatica, Via Diotisalvi 2, 56122 Pisa, Italy

Abstract

Nowadays the emphasis in software engineering research is on the evolution of pre-existing sub-systems and component development. In this context, we tackle the following problem: given the formal specification of the system P, already built, how to characterize possible collaborators of P, through a given communication interface L, to the satisfaction of a given property φ. We propose an abstract interpretation framework to reason about this problem in a systematic way. Given P and L, the set of all transition systems that, composed with P and restricted by L, satisfy φ, is modeled as the abstract semantics of φ, parametric with respect to P and L. We show that the algorithm developed by Andersen [1] can be formulated in our framework.

Key words: Compositional verification, Temporal logic, Concurrency

1. Introduction

Software has been evolving from pre-defined, monolithic architectures to dynamically composed federations of components. The needed software architectures provide flexibility, but also raise a number of challenges; in particular, verification becomes very hard. Compositional techniques for construction and analysis of software have shown to be effective tools for breaking down the complexity of systems construction and verification to manageable sizes.

*Corresponding author

Email addresses: nico@iet.unipi.it (N. De Francesco), g.lettieri@iet.unipi.it (G. Lettieri), luca.martini@iet.unipi.it (L. Martini), g.vaglini@iet.unipi.it (G. Vaglini)

Formal methods for verification are indispensable in the development of reliable software systems. Specification and verification of concurrent systems typically use automata or temporal logic as its foundation and automata based approaches quickly lead to state explosion in large systems (see, for example, the model checking approach [2]). Thus it is important that not only the construction of the system, but also its verification can be made using compositional techniques that allow the separate verification of the system components and avoid the construction of the automaton corresponding to the complete system. In this work, we have the specification of the system P and our aim is to allow P to have a correct interaction, through a given communication interface L, with some other process Q so their collaboration leads to the satisfaction of the property φ.

Here, we consider properties described by temporal logic formulas expressed through the mu-calculus [3] and systems specified by CCS processes [4]. Moreover, if Q must collaborate with P to satisfy φ, its structure is not relevant provided that Q satisfies another property ψ such that, when P and Q are put in parallel through the interface L, the whole process satisfies φ. Thus, starting from the formula φ, to be satisfied by the complete system, the aim is to identify a formula ψ such that, for each process Q satisfying ψ, $(P \parallel Q)\backslash L$ satisfies φ. The general problem of formula synthesis has been solved by Andersen in [1] through the partial model checking technique for the full mu-calculus. This technique, which is sound and complete, is proposed as a solution to the state explosion problem encountered when verifying a concurrent system with many parallel processes, say $(P_1 \parallel \ldots \parallel P_n)\backslash L$. Instead of model checking the entire process against φ, one can deduce the property that has to be verified by a sub-system, e.g., $(P_j \parallel \ldots \parallel P_n)$, taking into account $P_1 \parallel \ldots \parallel P_{j-1}$, L and φ (thus, as in our case, the technique applies to $(P_1 \parallel \ldots \parallel P_{j-1} \parallel X)\backslash L$ and φ). Clearly, this process can be iterated until reaching a component where model checking is feasible. Andersen's technique includes in ψ all the possible behaviors of $P_1 \parallel \ldots \parallel P_{j-1}$, so producing a formula whose complexity depends on the number of states of $P_1 \parallel \ldots \parallel P_{j-1}$.

We propose an abstract interpretation framework (see [5]) to reason about this problem in a systematic way. This framework is based on the work of Cousot and Cousot [6], where compositional abstract interpretations are developed. In the approach we identify the semantics of a formula ψ as the set of transition systems satisfying ψ. Given P and L, the set of all transition systems that, composed with P and restricted by L, satisfy φ, is modeled as the abstract semantics of φ, parametric with respect to P and L. The idea is that different approximations of the abstract semantics give rise to different algorithms. The soundness of each algorithm is ensured by the abstract interpretation framework. As an example, we show that the algorithm developed by Andersen [1] can be formulated in our framework. The missing proofs of the propositions stated in the article can be found in the companion technical report [7].

2. Background

2.1. Abstract interpretation

Let $\langle C, \leq \rangle$ and $\langle A, \sqsubseteq \rangle$ be posets and α, γ be maps from C to A and from A to C, respectively. If $\forall x \in C, y \in A$, $\alpha(x) \sqsubseteq y \iff x \leq \gamma(y)$, then we say that the pair (α, γ) is a *Galois connection*, and we write $\langle C, \leq \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle A, \sqsubseteq \rangle$. The map α is called the *upper adjoint*, while γ is the *lower adjoint*. We say that C is the *concrete* domain, and A is the *abstract* domain. We have that α preserves all existing joins, while γ preservers all existing meets. The image of C through α, i.e., $\alpha(C)$, is isomorphic to $\gamma(A)$, with α and γ being a pair of mutually inverse isomorphisms. The duality principle for Galois connections states that $\langle C, \leq \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle A, \sqsubseteq \rangle \iff \langle A, \sqsupseteq \rangle \underset{\gamma}{\overset{\alpha}{\leftrightarrows}} \langle C, \geq \rangle$.

Given a Galois connection $\langle C, \leq \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle A, \sqsubseteq \rangle$, all these three conditions are equivalent: (1) α is surjective; (2) γ is injective; (3) $\forall y \in A, (\alpha \circ \gamma)(y) = y$. When these conditions hold, we say that (α, γ) is a *Galois insertion* between $\langle C, \leq \rangle$ and $\langle A, \sqsubseteq \rangle$ and we write $\langle C, \leq \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows\!\!\!\!\rightarrow}} \langle A, \sqsubseteq \rangle$.

Given a Galois connection $\langle C, \leq \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle A, \sqsubseteq \rangle$, it is always possible to obtain a Galois insertion by identifying the elements in A with the same γ-image. That is, given the equivalence relation $\equiv \subseteq A \times A$ defined as $y_1 \equiv y_2 \iff \gamma(y_1) =$

$P, Q ::=$	processes	
	$a.P$	action
	X	variable
	$P + Q$	choice
	$P \parallel Q$	parallel composition
	$P\backslash L$	restriction
	$P[f]$	relabelling

Figure 1: Syntax of processes

$\gamma(y_2)$, we have $\langle C, \leq\rangle \underset{\alpha'}{\overset{\gamma'}{\leftrightarrows}} \langle A', \sqsubseteq'\rangle$, where $A' = A/_{\equiv}$, for all $[x]_{\equiv}, [y]_{\equiv} \in A/_{\equiv}$ we let $[x]_{\equiv} \sqsubseteq' [y]_{\equiv}$ iff $\gamma(x) \leq \gamma(y)$, and we define $\alpha'(x) = [\alpha(x)]_{\equiv}$ and $\gamma'([x]_{\equiv}) = \gamma(x)$. Note that $A/_{\equiv}$ is isomorphic to $\gamma'(A) = \gamma(A)$, which is isomorphic to $\alpha(C)$.

If $\langle C, \leq\rangle$ and $\langle A, \sqsubseteq\rangle$ are two Boolean lattices and $f\colon C \to A$, we define the dual of f as the function $\tilde{f}\colon C \to A$ given by $\tilde{f}(x) = \neg f(x')$ for all $x \in C$, where $\neg$ denotes complementation in A and x' is the complement of x in C. If $\langle C, \leq\rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle A, \sqsubseteq\rangle$, then $\langle A, \sqsubseteq\rangle \underset{\tilde{\gamma}}{\overset{\tilde{\alpha}}{\leftrightarrows}} \langle C, \leq\rangle$.

2.2. Process algebra and CCS

$$\textbf{Act}\ \frac{}{a.P \xrightarrow{a} P} \qquad \textbf{Res}\ \frac{P \xrightarrow{a} P'}{P\backslash L \xrightarrow{a} P'\backslash L}\ a, \overline{a} \notin L$$

$$\textbf{Sum}\ \frac{P \xrightarrow{a} P'}{P + Q \xrightarrow{a} P'} \qquad \textbf{Con}\ \frac{P \xrightarrow{a} P'}{X \xrightarrow{a} P'}\ X \triangleq P$$

$$\textbf{Par1}\ \frac{P \xrightarrow{a} P'}{P \parallel Q \xrightarrow{a} P' \parallel Q} \qquad \textbf{Par2}\ \frac{Q \xrightarrow{a} Q'}{P \parallel Q \xrightarrow{a} P \parallel Q'}$$

$$\textbf{Rel}\ \frac{P \xrightarrow{a} P'}{P[f] \xrightarrow{f(a)} P'[f]} \qquad \textbf{Com}\ \frac{P \xrightarrow{a} P', Q \xrightarrow{\overline{a}} Q'}{P \parallel Q \xrightarrow{\tau} P' \parallel Q'}$$

Figure 2: Operational semantics

Processes are described by the syntax in Figure 1. Process semantics is described in an operational fashion by the rules of Figure 2. The process $P + Q$ is equivalent to the process $Q + P$. The complement of an action a is the action $\bar{a}$. We suppose that the set of actions $\mathbb{A}$ is closed under $^-$ and $\overline{\overline{a}} = a$. The special

$\varphi, \psi ::=$	formulas
$\mathtt{tt}$	true
$\mathtt{ff}$	false
X	variable
$\varphi \wedge \psi$	conjunction
$\varphi \vee \psi$	disjunction
$\langle a \rangle \varphi$	diamond
$[a]\varphi$	box
$\nu X.\varphi$	greatest fixpoint
$\mu X.\varphi$	least fixpoint

Figure 3: Syntax of formulas

Φ	ranged over by $\varphi, \psi, \ldots$	formulas
$\mathbb{V}$	ranged over by $X, Y, \ldots$	variables
$\mathbb{P}$	ranged over by $P, Q, \ldots$	processes
$\mathbb{A}$	ranged over by $a, b, \ldots$	actions
$\mathbb{S}$	ranged over by $s, s', \ldots$	states
$\mathbb{T}$	ranged over by $t, t', \ldots$	transition systems
$\mathbb{E}$	ranged over by $\sigma, \rho, \ldots$	environments

Figure 4: Domains

label $\tau \notin \mathbb{A}$ denotes the silent action. In the following, we write $\mathbb{A}_\tau$ in place of $\mathbb{A} \cup \{\tau\}$.

A process can also be represented equivalently by its transition system. A transition system $t \in \mathbb{T}$ is a triple $\langle S, \rightarrow, s_0 \rangle$, where $S \subseteq \mathbb{S}$, is the set of states, $\rightarrow \subseteq S \times \mathbb{A}_\tau \times S$ is the relation among states that produces actions, and $s_0 \in S$ is the initial state.

Following [1], we define the parallel composition and the restriction operator of Figure 2 directly in terms of transition systems. The composition $\langle S_1, \rightarrow_1, s_1^0 \rangle \parallel \langle S_2, \rightarrow_2, s_2^0 \rangle$ is the transition system $\langle S_1 \times S_2, \rightarrow, (s_1^0, s_2^0) \rangle$ with the relation $\rightarrow$ defined as:

$$\begin{aligned}(s_1, s_2) \xrightarrow{a} (s_1', s_2') \iff & (s_1 = s_1' \text{ and } s_2 \xrightarrow{a}_2 s_2') \text{ or} \\ & (s_1 \xrightarrow{a}_1 s_1' \text{ and } s_2 = s_2') \text{ or} \\ & (a = \tau \text{ and } \exists b : s_1 \xrightarrow{b}_1 s_1' \text{ and } s_2 \xrightarrow{\bar{b}}_2 s_2').\end{aligned}$$

Analogously, the transition system $\langle S, \rightarrow, s_0 \rangle \backslash L$ is the transition system $\langle S, \rightarrow_L, s_0 \rangle$

$[\![\mathtt{tt}]\!]\rho = \mathbb{T}$ $\quad [\![\mathtt{ff}]\!]\rho = \varnothing$	$[\![X]\!]\rho = \rho(X)$
$[\![\varphi \wedge \psi]\!]\rho = [\![\varphi]\!]\rho \cap [\![\psi]\!]\rho$	$[\![\varphi \vee \psi]\!]\rho = [\![\varphi]\!]\rho \cup [\![\psi]\!]\rho$
$[\![\langle a \rangle \varphi]\!]\rho = \mathcal{B}_a([\![\varphi]\!]\rho)$	$[\![\mu X.\varphi]\!]\rho = \mathsf{lfp}\, \lambda x.[\![\varphi]\!](\rho\,[x/X])$
$[\![[a]\varphi]\!]\rho = \tilde{\mathcal{B}}_a([\![\varphi]\!]\rho)$	$[\![\nu X.\varphi]\!]\rho = \mathsf{gfp}\, \lambda x.[\![\varphi]\!](\rho\,[x/X])$

Figure 5: Formula semantics

with the relation $\rightarrow_L$ defined as follows:

$$s \xrightarrow{a}_L s' \iff s \xrightarrow{a} s' \text{ and } a \notin L.$$

In the following, we will extend the operators of parallel composition and restriction to sets of transition systems, that is, given $T_1, T_2 \in \wp(\mathbb{T})$, and $L \subseteq \mathbb{A}$,

$$\begin{aligned} T_1 \parallel T_2 &= \{t_1 \parallel t_2 \mid t_1 \in T_1, t_2 \in T_2\}, \\ T_1 \backslash L &= \{t \backslash L \mid t \in T_1\}. \end{aligned}$$

To reduce the number of parentheses, we suppose that these operators take precedence over the set union and set intersection operators.

Formulas syntax is described in Figure 3. We consider finite formulas. Formulas may contain variables. An environment $\rho \colon \mathbb{V} \to \wp(\mathbb{T})$ assigns values to the free variables of a formula. Let $\mathbb{E} = \mathbb{V} \to \wp(\mathbb{T})$ be the domain of environments. In the following, we will denote with $f\,[z/y]$ the function f' such that $f'(x) = f(x)$ when $x \neq y$, and $f'(x) = z$ otherwise. The semantics of a formula is defined by a function $[\![\bullet]\!] \colon \Phi \to \mathbb{E} \to \wp(\mathbb{T})$. The definition of $[\![\bullet]\!]$ is in Figure 5, where $\mathcal{B}$ and $\tilde{\mathcal{B}}$ are functions from $\wp(\mathbb{T})$ to $\wp(\mathbb{T})$ defined as

$$\mathcal{B}_a(T) = \{\langle S, \rightarrow, s\rangle \mid \exists s' : s \xrightarrow{a} s' \text{ and } \langle S, \rightarrow, s'\rangle \in T\}, \tag{1}$$

$$\tilde{\mathcal{B}}_a(T) = \{\langle S, \rightarrow, s\rangle \mid \forall s',\ s \xrightarrow{a} s' \implies \langle S, \rightarrow, s'\rangle \in T\}. \tag{2}$$

Please note that $\tilde{\mathcal{B}}_a$ is the dual function of $\mathcal{B}_a$.

When a transition system t belongs to $[\![\varphi]\!]\rho$ we will say that t *satisfies* φ within the environment ρ, and we will write $t \models_\rho \varphi$. When the environment is empty, or clear from the context, we will omit the subscript ρ.

3. The problem

Given a process P and a formula φ and a set of actions L, consider the problem of finding a process Q, such that

$$(P \parallel Q)\backslash L \models \varphi. \tag{3}$$

Since there can exist more than one process Q that satisfy (3), we are interested in finding a formula ψ, such that if $Q \models \psi$, then also (3) holds.

We can define the following *checking abstraction* $\alpha_{P,L} \colon \wp(\mathbb{T}) \to \wp(\mathbb{T})$ and its corresponding concretization $\gamma_{P,L} \colon \wp(\mathbb{T}) \to \wp(\mathbb{T})$ in this way:

$$\alpha_{P,L}(T) = \{\, t \mid (P \parallel t)\backslash L \in T \,\}, \tag{4}$$

$$\gamma_{P,L}(T) = \{\, (P \parallel t)\backslash L \mid t \in T \,\}. \tag{5}$$

Proposition 3.1. *Let $P \in \mathbb{P}$, and $L \subseteq \mathbb{A}$. Then, $\langle \wp(\mathbb{T}), \supseteq \rangle \underset{\alpha_{P,L}}{\overset{\gamma_{P,L}}{\leftrightarrows}} \langle \wp(\mathbb{T}), \supseteq \rangle$.*

Proof. By functional abstraction we have that $\langle \wp(\mathbb{T}), \subseteq \rangle \underset{\gamma_{P,L}}{\overset{\alpha_{P,L}}{\leftrightarrows}} \langle \wp(\mathbb{T}), \subseteq \rangle$. We can derive the thesis by applying the duality principle. □

Proposition 3.2. *For all $P \in \mathbb{P}$ and $L \subseteq \mathbb{A}$, function $\alpha_{P,L}$ defined as in* (4) *is self-dual, i.e., $\tilde{\alpha}_{P,L} = \alpha_{P,L}$.*

Proof. For all $T \in \wp(\mathbb{T})$ and $t \in \mathbb{T}$ we have $t \in \tilde{\alpha}_{P,L}(T) \iff t \notin \{\, t' \mid (P \parallel t')\backslash L \notin T \,\} \iff t \in \{\, t' \mid (P \parallel t')\backslash L \in T \,\} \iff t \in \alpha_{P,L}(T)$. □

Prop. 3.2 implies that $\alpha_{P,L}$ participates in two Galois connections between $\langle \wp(\mathbb{T}), \subseteq \rangle$ and itself, once as an upper and once as a lower adjoint. Thus, $\alpha_{P,L}$ preserves arbitrary intersections and unions. This is immediately clear if we regard $(P \parallel t)\backslash L$ as a function $f(t) \colon \mathbb{T} \to \mathbb{T}$. Then $\alpha_{P,L}(T)$ is simply the inverse image of T through f.

Please note that, in general, $(\alpha_{P,L}, \gamma_{P,L})$ is not a Galois insertion between $\langle \wp(\mathbb{T}), \supseteq \rangle$ and itself. In fact, $\gamma_{P,L}$ may not be injective. For instance, if $L \neq \varnothing$, then all sets containing only transition systems that can only do actions in L and that do not communicate with P have the same $\gamma_{P,L}$-image. Therefore, we reduce the Galois connection to a Galois insertion in the canonical way, by defining the equivalence relation $\approx_{P,L} \subseteq \wp(\mathbb{T}) \times \wp(\mathbb{T})$ such that

$$T_1 \approx_{P,L} T_2 \iff \gamma_{P,L}(T_1) = (P \parallel T_1)\backslash L = (P \parallel T_2)\backslash L = \gamma_{P,L}(T_2). \tag{6}$$

Then we restrict the abstract domain to

$$\mathscr{F}_{P,L} = \{\, \alpha_{P,L}(T) \mid T \in \wp(\mathbb{T}) \,\} = \alpha_{P,L}(\wp(\mathbb{T})), \tag{7}$$

which is isomorphic to $\wp(\mathbb{T})/_{\approx_{P,L}}$, and obtain $\langle \wp(\mathbb{T}), \supseteq \rangle \underset{\alpha_{P,L}}{\overset{\gamma_{P,L}}{\leftrightarrows}} \langle \mathscr{F}_{P,L}, \supseteq \rangle$. Since $\alpha_{P,L}$ is a surjection that preserves all existing joins and meets and $\wp(\mathbb{T})$ is a complete Boolean lattice, domain $\mathscr{F}_{P,L}$ is a complete Boolean lattice too. The elements $T \in \mathscr{F}_{P,L}$ are characterized by the fact that, whenever there exist $t \in T$ and $s \in \mathbb{T}$ such that $(P \parallel t)\backslash L = (P \parallel s)\backslash L$, then $s \in T$. Figure 7(a) outlines our approach. Given an environment $\rho \in \mathbb{E}$, we would like to obtain the set of transition systems $T^{\#}$ such that $\forall t \in T^{\#}$, $(P \parallel t)\backslash L \models_{\rho} \varphi$. This is precisely the set $\alpha_{P,L}([\![\varphi]\!]\rho)$. We want to compute this set in a compositional way depending on the syntax of φ. Since modal operators force us to consider process P with different initial states, we also have to compute $\alpha_{P',L}([\![\varphi]\!]\rho)$, with P' reachable from P. We organize all these sets in a vector in the following way. Let $\mathcal{P}$ be the (finite) set of all processes reachable from P. We introduce $\overline{\alpha}\colon \wp(\mathbb{T}) \to \Pi_{Q \in \mathcal{P}} \mathscr{F}_{Q,L}$ given by

$$(\overline{\alpha}(T))_Q = \alpha_{Q,L}(T), \tag{8}$$

for all $T \in \wp(\mathbb{T})$ and $Q \in \mathcal{P}$. Note that we have denoted with $(v)_Q$ the component of vector v whose index is Q.

We now want to define vector abstraction $[\![\varphi]\!]^{\#}\colon (\mathbb{V} \to \Pi_{Q \in \mathcal{P}} \mathscr{F}_{Q,L}) \to \Pi_{Q \in \mathcal{P}} \mathscr{F}_{Q,L}$ such that, for each $\rho \in \mathbb{E}$,

$$\overline{\alpha}([\![\varphi]\!]\rho) = [\![\varphi]\!]^{\#}(\overline{\alpha} \circ \rho). \tag{9}$$

If Equation (9) can be satisfied by a proper definition of $[\![\bullet]\!]^{\#}$, then the element of vector $[\![\varphi]\!]^{\#}\overline{\alpha} \circ \rho$ indexed by P is the solution we are looking for. Note that for formulas with no free variables, ρ can be taken to be $\lambda X.\varnothing$ and $\overline{\alpha} \circ \rho$ requires no computation.

Before describing the abstract semantics function $[\![\bullet]\!]^{\#}$, we must introduce some definitions that are useful for dealing with the temporal logic formulas.

For each $a \in \mathbb{A}_\tau$, we define a *forward* function $\mathcal{F}_a \colon \wp(\mathbb{T}) \to \wp(\mathbb{T})$ such that

$$\mathcal{F}_a(T) = \{\langle S, \to, s' \rangle \mid \exists \langle S, \to, s \rangle \in T : s \xrightarrow{a} s'\}. \tag{10}$$

The set $\mathcal{F}_a(T)$ contains all the processes in T, after an action a. If t is a process, we write $\mathcal{F}_a(t)$ as a shorthand for $\mathcal{F}_a(\{t\})$. Following the Cousot work, we can now state the following proposition.

Proposition 3.3. *Given $a \in \mathbb{A}_\tau$, we have $\langle \wp(\mathbb{T}), \subseteq \rangle \underset{\mathcal{F}_a}{\overset{\tilde{\mathcal{B}}_a}{\leftrightarrows}} \langle \wp(\mathbb{T}), \subseteq \rangle$.*

By combining equation 10 with the definition of parallel composition of transition systems, we obtain the following proposition.

Proposition 3.4. *Given $T, T_1, T_2 \in \wp(\mathbb{T})$, $a \in \mathbb{A}_\tau$ and $L \subseteq \mathbb{A}$, we have*

$$\mathcal{F}_a(T_1 \parallel T_2) = \begin{cases} \mathcal{F}_a(T_1) \parallel T_2 \cup T_1 \parallel \mathcal{F}_a(T_2) & \text{if } a \neq \tau, \\ \mathcal{F}_\tau(T_1) \parallel T_2 \cup T_1 \parallel \mathcal{F}_\tau(T_2) \cup \bigcup\limits_{b \in \mathbb{A}} \mathcal{F}_b(T_1) \parallel \mathcal{F}_{\bar{b}}(T_2) & \text{if } a = \tau, \end{cases} \tag{11}$$

$$\mathcal{F}_a(T \backslash L) = \begin{cases} \emptyset & \text{if } a \in L, \\ \mathcal{F}_a(T) \backslash L & \text{if } a \notin L. \end{cases} \tag{12}$$

We now have all the tools needed to prove the following proposition.

Proposition 3.5. *The vector abstraction defined in Figure 6 satisfies* (9) *for all $\rho \in \mathbb{E}$.*

Proof. By structural induction on the syntax of formula φ. Let Q be any process reachable from P.

If $\varphi = \mathtt{tt}$, then $(\overline{\alpha}([\![\mathtt{tt}]\!]\rho))_Q = \alpha_{Q,L}(\mathbb{T}) = \mathbb{T} = ([\![\mathtt{tt}]\!]^{\#}\overline{\alpha} \circ \rho)_Q$.

If $\varphi = X$, then $(\overline{\alpha}([\![X]\!]\rho))_Q = \alpha_{Q,L}(\rho(X)) = (\overline{\alpha} \circ \rho(X))_Q = ([\![X]\!]^{\#}\overline{\alpha} \circ \rho)_Q$.

If $\varphi = \varphi_1 \wedge \varphi_2$, then $(\overline{\alpha}([\![\varphi_1 \wedge \varphi_2]\!]\rho))_Q = \alpha_{Q,L}([\![\varphi_1]\!]\rho \cap [\![\varphi_2]\!]\rho) =$ [by Prop. 3.2] $\alpha_{Q,L}([\![\varphi_1]\!]\rho) \cap \alpha_{Q,L}([\![\varphi_2]\!]\rho) = (\overline{\alpha}([\![\varphi_1]\!]\rho))_Q \cap (\overline{\alpha}([\![\varphi_2]\!]\rho))_Q =$ [by induction hypothesis] $([\![\varphi_1]\!]^{\#}\overline{\alpha} \circ \rho)_Q \cap ([\![\varphi_2]\!]^{\#}\overline{\alpha} \circ \rho)_Q = ([\![\varphi_1 \wedge \varphi_2]\!]^{\#}\overline{\alpha} \circ \rho)_Q$.

If $\varphi = [a]\varphi_1$, then $(\overline{\alpha}([\![[a]\varphi_1]\!]\rho))_Q = \alpha_{Q,L}(\tilde{\mathcal{B}}_a([\![\varphi_1]\!]\rho)) =$ [by definition of $\alpha_{Q,L}$] $\{\, t \mid (Q \parallel t)\backslash L \in \tilde{\mathcal{B}}_a([\![\varphi_1]\!]\rho) \,\} =$ [by Proposition 3.3] $\{\, t \mid \mathcal{F}_a((Q \parallel t)\backslash L) \subseteq [\![\varphi_1]\!]\rho \,\}$. We now apply Equation (11) and (12). We have three cases, depending on the value of a. If $a \in L$, then $(\overline{\alpha}([\![[a]\varphi_1]\!]\rho))_Q = \{\, t \mid \varnothing \subseteq [\![\varphi_1]\!]\rho \,\} = \mathbb{T} = ([\![\varphi]\!]^{\#}\overline{\alpha} \circ \rho)_Q$. If $a \notin L$ and $a \neq \tau$, then $(\overline{\alpha}([\![[a]\varphi_1]\!]\rho)_Q = \{\, t \mid (\mathcal{F}_a(Q) \parallel t)\backslash L \cup (Q \parallel \mathcal{F}_a(t))\backslash L \subseteq [\![\varphi_1]\!]\rho \,\} = \{\, t \mid (\mathcal{F}_a(Q) \parallel t)\backslash L \subseteq [\![\varphi_1]\!]\rho \,\} \cap \{\, t \mid (Q \parallel \mathcal{F}_a(t))\backslash L \subseteq [\![\varphi_1]\!]\rho \,\} = \bigcap_{Q' \in \mathcal{F}_a(Q)} \{\, t \mid (Q' \parallel t)\backslash L \in [\![\varphi_1]\!]\rho \,\} \cap \{\, t \mid \mathcal{F}_a(t) \subseteq \alpha_{Q,L}([\![\varphi_1]\!]\rho) \,\} = \bigcap_{Q' \in \mathcal{F}_a(Q)} \alpha_{Q',L}([\![\varphi_1]\!]\rho) \cap \{\, t \mid \mathcal{F}_a(t) \subseteq \alpha_{Q,L}([\![\varphi_1]\!]\rho) \,\} = \bigcap_{Q' \in \mathcal{F}_a(Q)} ([\![\varphi_1]\!]^{\#}\overline{\alpha} \circ \rho)_{Q'} \cap \{\, t \mid \mathcal{F}_a(t) \subseteq ([\![\varphi_1]\!]^{\#}\overline{\alpha} \circ \rho)_Q \,\} =$ [by Proposition 3.3] $\bigcap_{Q' \in \mathcal{F}_a(Q)} ([\![\varphi_1]\!]^{\#}\overline{\alpha} \circ \rho)_{Q'} \cap \tilde{\mathcal{B}}_a([\![\varphi_1]\!]^{\#}\overline{\alpha} \circ \rho)_Q = ([\![[a]\varphi_1]\!]^{\#}\overline{\alpha} \circ \rho)_Q$. If $a = \tau$, then $(\overline{\alpha}([\![[a]\varphi_1]\!]\rho)_Q =$

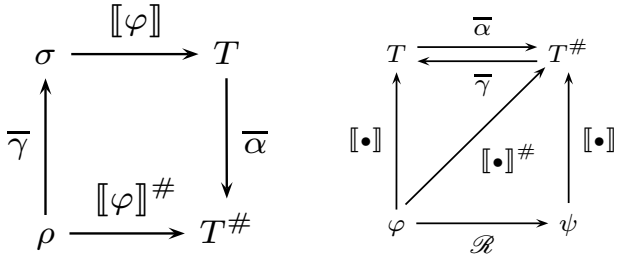
$$
\begin{aligned}
[\![\mathtt{tt}]\!]^{\#}\sigma &= \overline{\mathbb{T}} \\
[\![\mathtt{ff}]\!]^{\#}\sigma &= \overline{\varnothing} \\
[\![X]\!]^{\#}\sigma &= \sigma(X) \\
([\![\varphi_1 \wedge \varphi_2]\!]^{\#}\sigma)_Q &= ([\![\varphi_1]\!]^{\#}\sigma)_Q \cap ([\![\varphi_2]\!]^{\#}\sigma)_Q \\
([\![\varphi_1 \vee \varphi_2]\!]^{\#}\sigma)_Q &= ([\![\varphi_1]\!]^{\#}\sigma)_Q \cup ([\![\varphi_2]\!]^{\#}\sigma)_Q \\
([\![[a]\varphi]\!]^{\#}\sigma)_Q &= \begin{cases} \mathbb{T} & a \in L \\ \bigcap_{Q' \in \mathcal{F}_a(Q)} ([\![\varphi]\!]^{\#}\sigma)_{Q'} \cap \tilde{\mathcal{B}}_a([\![\varphi]\!]^{\#}\sigma)_Q & a \notin L, a \neq \tau \\ \bigcap_{Q' \in \mathcal{F}_\tau(Q)} ([\![\varphi]\!]^{\#}\sigma)_{Q'} \cap \tilde{\mathcal{B}}_\tau([\![\varphi]\!]^{\#}\sigma)_Q \cap \bigcap_{b \in \mathbb{A}, Q' \in \mathcal{F}_b(Q)} \tilde{\mathcal{B}}_{\bar{b}}([\![\varphi]\!]^{\#}\sigma)_{Q'} & a = \tau \end{cases} \\
([\![\langle a\rangle\varphi]\!]^{\#}\sigma)_Q &= \begin{cases} \varnothing & a \in L \\ \bigcup_{Q' \in \mathcal{F}_a(Q)} ([\![\varphi]\!]^{\#}\sigma)_{Q'} \cup \mathcal{B}_a([\![\varphi]\!]^{\#}\sigma)_Q & a \notin L, a \neq \tau \\ \bigcup_{Q' \in \mathcal{F}_\tau(Q)} ([\![\varphi]\!]^{\#}\sigma)_{Q'} \cup \mathcal{B}_\tau([\![\varphi]\!]^{\#}\sigma)_Q \cup \bigcup_{b \in \mathbb{A}, Q' \in \mathcal{F}_b(Q)} \mathcal{B}_{\bar{b}}([\![\varphi]\!]^{\#}\sigma)_{Q'} & a = \tau \end{cases} \\
[\![\mu X.\varphi]\!]^{\#}\sigma &= \mathsf{lfp}\, \lambda\overline{x}.[\![\varphi]\!]^{\#}\sigma\,[\overline{x}/X] \\
[\![\nu X.\varphi]\!]^{\#}\sigma &= \mathsf{gfp}\, \lambda\overline{x}.[\![\varphi]\!]^{\#}\sigma\,[\overline{x}/X]
\end{aligned}
$$

Figure 6: Formula semantics

$\{\, t \mid (\mathcal{F}_\tau(Q) \parallel t)\backslash L \cup (Q \parallel \mathcal{F}_\tau(t))\backslash L \cup \bigcup_{b\in\mathbb{A}}(\mathcal{F}_b(Q) \parallel \mathcal{F}_{\bar{b}}(t))\backslash L \subseteq [\![\varphi_1]\!]\rho \,\} = \bigcap_{Q'\in\mathcal{F}_\tau(Q)}([\![\varphi_1]\!]^{\#}\overline{\alpha}\circ\rho)_{Q'} \cap \tilde{\mathcal{B}}_\tau([\![\varphi_1]\!]^{\#}\rho)_Q \cap \bigcap_{b\in\mathbb{A},Q'\in\mathcal{F}_b(Q)}\{\, t \mid (Q' \parallel \mathcal{F}_{\bar{b}}(t))\backslash L \subseteq [\![\varphi_1]\!]\rho \,\} = \bigcap_{Q'\in\mathcal{F}_\tau(Q)}([\![\varphi_1]\!]^{\#}\overline{\alpha}\circ\rho)_{Q'} \cap \tilde{\mathcal{B}}_\tau([\![\varphi_1]\!]^{\#}\rho)_Q \cap \bigcap_{b\in\mathbb{A},Q'\in\mathcal{F}_b(Q)} \tilde{\mathcal{B}}_{\bar{b}}([\![\varphi_1]\!]^{\#}\overline{\alpha}\circ\rho)_{Q'} = ([\![[a]\varphi_1]\!]^{\#}\overline{\alpha}\circ\rho)_Q$.

If $\varphi = \mu X.\varphi_1$, let $F = \lambda x.[\![\varphi_1]\!]\rho\,[x/X]$ and $F^{\#} = \lambda\overline{x}.[\![\varphi_1]\!]^{\#}(\overline{\alpha}\circ\rho)\,[\overline{x}/X]$. We claim that $\overline{\alpha}\circ F = F^{\#}\circ\overline{\alpha}$. In fact, for all $x \in \wp(\mathbb{T})$, we can use the induction hypothesis and obtain $\overline{\alpha}(F(x)) = \overline{\alpha}([\![\varphi_1]\!]\rho\,[x/X]) = [\![\varphi_1]\!]^{\#}\overline{\alpha}\circ(\rho\,[x/X]) = [\![\varphi_1]\!]^{\#}(\overline{\alpha}\circ\rho)\,[\overline{\alpha}(x)/X] = F^{\sharp}(\overline{\alpha}(x))$. Then, for the Fixpoint Fusion theorem, we get $\overline{\alpha}([\![\mu X.\varphi_1]\!]\rho) = \overline{\alpha}(\mathsf{lfp}\,F) = \mathsf{lfp}\,F^{\sharp} = [\![\mu X.\varphi_1]\!]^{\#}\overline{\alpha}\circ\rho$. The other cases are dual of the previous. □

Equation 9 is the usual *completeness* condition of an abstract interpretation. It is also easy to prove that $[\![\varphi]\!]^{\#}$ is the *best abstraction* of $[\![\varphi]\!]$. When the cost of calculating $[\![\varphi]\!]^{\#}$ is prohibitive one may think to use another (simpler) interpretation of $[\![\varphi]\!]$, possibly releasing the completeness constraint but always preserving soundness. In this way the set of transition systems calculated by this interpretation will be a subset of that calculated by $[\![\varphi]\!]^{\#}$. Comparing the definition of $[\![\varphi]\!]^{\#}$ with the semantics of formulas (Figure 5) it is not difficult to obtain the result of Andersen [1]. In fact, we can define a function $\mathscr{R}$: $(\Phi \times$



(a) $[\![\bullet]\!]^{\#}$ as the best abstraction

(b) Partial model checking, from φ to ψ

Figure 7: Commutative diagrams

$\mathbb{P}) \to \Phi$ starting from the abstract semantics $[\![\bullet]\!]^{\#}_{Q}$ and substituting set union (intersection) with logic disjunction (conjunction), and the function $\mathcal{B}$ ($\tilde{\mathcal{B}}$) with the diamond (box) operator. As an example, if $a \notin L, a \neq \tau$,

$$\mathscr{R}_Q([a]\varphi) = \bigwedge_{Q' \in \mathcal{F}_a(Q)} \mathscr{R}_{Q'}(\varphi) \wedge [a]\mathscr{R}_Q(\varphi).$$

Then we can prove that $[\![\mathscr{R}_P(\varphi)]\!] = [\![\varphi]\!]^{\#}_{P}$ by structural induction on φ. We can now prove the following proposition:

Proposition 3.6 (Partial model checking)**.** *Given two processes $P, Q \in \mathbb{P}$, a formula $\varphi \in \Phi$ with no free variables, and a set of actions $L \in \mathbb{A}$, we have that*

$$Q \models \mathscr{R}_P(\varphi) \iff (P \parallel Q)\backslash L \models \varphi \tag{13}$$

Proof. This proof can be graphically represented by Figure 7(b). □

4. Conclusions

In this work we applied abstract interpretation techniques to recover the description of the processes (either by means of a formula or by means of its transition system) that, composed with a known process through a given communication interface, lead to the satisfaction of a given property. The abstract interpretation framework allowed us to prove a completeness result in a quasi-constructive way.

References

[1] H. R. Andersen, Partial model checking (extended abstract), in: Proceedings of the Tenth Annual IEEE Symposium on Logic in Computer Science, IEEE Computer Society Press, 1995, pp. 398–407.

[2] E. M. Clarke, O. Grumberg, D. Peled, "Model checking", MIT Press, 2000.

[3] C. Stirling, An introduction to modal and temporal logics for CCS, in: Concurrency: Theory, Language, And Architecture, LNCS, 1989, pp. 2–20.

[4] R. Milner, Communication and Concurrency, Prentice Hall (International Series in Computer Science), 1989.

[5] P. Cousot, R. Cousot, Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints, in: 4th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages Proceedings, Los Angeles, California, 1977, pp. 238–252.

[6] P. Cousot, R. Cousot, Temporal abstract interpretation, in: Conference Record of the Twentyseventh Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, ACM Press, New York, NY, Boston, Mass., 2000, pp. 12–25.

[7] N. D. Francesco, G. Lettieri, L. Martini, G. Vaglini, Partial model checking via abstract interpretation, Tech. Rep. IET-09-09, Dipartimento di Ingegneria dell'Informazione, Università di Pisa (2009).
URL `http://calcolatori.iet.unipi.it/IET-09-09.pdf`