

Comparing Under and Over-Approximations of LTL Properties for Model Checking

María del Mar Gallardo ^{a,1,2} Pedro Merino ^{a,3}
Ernesto Pimentel ^{a,4}

^a *Dpto. de Lenguajes y Ciencias de la Computacion*
University of Malaga
29071 Malaga
Spain

Abstract

The classic method for abstracting temporal properties when realizing abstract model checking is based on defining an abstract satisfiability relation which under-approximates the standard one. As a consequence, satisfiability of universal properties is directly preserved from the abstract model to the concrete one. However, this result may be impractical due to the imprecision and incompleteness with which abstract models are usually constructed. Thus, in the case a model checking tool supporting abstract model checking gives a negative answer, the user must analyze the counter-examples produced to decide whether the property really fails or, on the contrary, the abstract model is too imprecise to obtain a definitive result. We have developed an alternative method for abstracting temporal properties based on the idea of over-approximation. In this paper, we compare these two methods with respect to the satisfiability/refutation of universal/existential properties, proving that they produce complementary results. Finally, we study the conditions which ensure that the method based on over-approximation also produces definitive answers when analyzing universal properties.

1 Introduction

Model Checking [1] represents one of the most useful results of almost twenty years of research in formal methods to increase the quality of software and other related systems. A model checker works with a high level description of

¹ This research is partially supported by the CICYT projects TIC2001-2705-C03-02 and TIC99-1083-C02-01.

² Email: gallardo@lcc.uma.es

³ Email: pedro@lcc.uma.es

⁴ Email: ernesto@lcc.uma.es

a system, a *model*, and it can automatically inspect the reachable states of the system to check if a given *property* is satisfied. Typically, the properties are expressed with some variant of temporal logic, where Linear time Temporal Logic (LTL) is one of the most employed [12].

In the context of model checking, abstract interpretation [2] is used as a way of dealing with the so-called *state explosion problem* which occurs when realistic systems are analyzed. Abstract model checking involves two activities. On the one hand, in order to reduce the state space of the original model M , we apply abstract interpretation to construct an abstract model M^α approximating M . On the other, we abstract the original satisfiability relation \models , which evaluates temporal properties against concrete models, and define an abstract relation \models^α to reinterpret the meaning of properties against the abstract models.

Given a generic temporal property f , the final objective of the abstraction process is the “strong preservation” (that is, the preservation of both the truth and the falsehood) of the universal ($\forall f$) and existential ($\exists f$) properties between M and M^α , in other words,

$$\begin{aligned} M^\alpha \models^\alpha \forall f &\Rightarrow M \models \forall f \quad (U1) & M^\alpha \models^\alpha \forall f &\Leftarrow M \models \forall f \quad (U2) \\ M^\alpha \models^\alpha \exists f &\Rightarrow M \models \exists f \quad (E1) & M^\alpha \models^\alpha \exists f &\Leftarrow M \models \exists f \quad (E2) \end{aligned}$$

However, the strong preservation of universal and existential properties is only possible if M and M^α are bi-similar [10], which entails a considerable constraint when the objective is to decrease the state space. Thus, it is accepted that a reasonable construction of abstract models may involve some loss of information when analyzing temporal properties.

The classic method [3,5] to abstract \models under-approximates properties in such a way that the abstract model satisfies less properties than the concrete one. This definition directly produces the weak preservation (U1) of universal properties. However, the remaining preservation results may fail due to the incompleteness/imprecision of the abstract model.

The way to tackle this problem is to analyze the counter-examples produced by the analysis to determine whether they are “spurious” as in [4] and [11] or to carry out a property-driven/domain-driven refinement of the abstract model as in [5].

In [5], the authors also present a proposal to analyze the existential properties (E1 and E2), which is based on the construction of a different abstract model M_e^α over which they are directly preserved.

We employ an *over-approximation* method developed in [8] for defining an abstract satisfiability relation that over-approximates properties. In this case, the abstract model satisfies more properties than the concrete one and the refutation of existential properties (E2) is now directly preserved. However, as in the classic method, we need to make an additional analysis in order to

achieve the remaining preservation results. This paper is devoted to comparing these two complementary methods. We also study how imprecision and incompleteness affect preservation results and state the conditions which guarantee that the over-approximation method may also be used to analyze the satisfiability of universal properties ($U1$).

Finally, we argue that a mixed method integrating the classic and the over-approximation approaches may be useful to improve the set of properties to be analyzed without modifying the abstract model. However, the description of this mixed method is beyond the aim of this paper, which is devoted to compare both approaches.

The paper is organized as follows. Section 2 is devoted to the construction of the abstract model. In Section 3, we present the classic and the over-approximation methods for abstracting temporal logic. We illustrate both methods with an example to show the natural way of applying each one. Section 4 discusses the problems of incompleteness and imprecision and finally, in Section 5, we give the conclusions.

2 Abstracting Concurrent Systems

Execution of a concurrent program may be defined by means of *labeled transition systems* (LTS) such as $M = (A, \Sigma, \overset{\rightarrow}{\rightarrow}, s_0)$ where A is the set of *observable atomic actions*, Σ is the set of standard *states*, $\overset{\rightarrow}{\rightarrow} \subseteq \Sigma \times A \times \Sigma$ is a labeled transition relation and s_0 is the initial state. We write $s \xrightarrow{a} s'$ for $(s, a, s') \in \overset{\rightarrow}{\rightarrow}$. A *trace* $x = t_0 \xrightarrow{a_0} t_1 \xrightarrow{a_1} \dots$ of M is a sequence of states and it represents a (possibly infinite) computation from state t_0 where $a_0 a_1 \dots$ is the sequence of atomic actions executed. Given a trace $x = t_0 \xrightarrow{a_0} t_1 \xrightarrow{a_1} \dots$, with x^j we will denote the suffix path $t_j \xrightarrow{a_j} t_{j+1} \xrightarrow{a_{j+1}} \dots$. A *full-trace* $x = t_0 \xrightarrow{a_0} \dots$ is a trace that cannot be extended in the future. We assume that terminating traces has a final state which is repeated forever. The set $\mathcal{O}(M) = \{x \mid x = s_0 \xrightarrow{a_0} \dots \text{ is a full-trace}\}$ defines the trace semantics determined by the transition system M .

An abstract interpretation $M^\alpha = (A, \Sigma^\alpha, \overset{\rightarrow}{\rightarrow}_\alpha, s_0^\alpha)$ of a labeled transition system $M = (A, \Sigma, \overset{\rightarrow}{\rightarrow}, s_0)$ is constructed by means of a triple $\mathcal{I}_\alpha = (\Sigma, (\Sigma^\alpha, \leq^\alpha), \beta)$. $(\Sigma^\alpha, \leq^\alpha)$ is a lattice of abstract states where partial order \leq^α represents the degree of precision of each abstract state, the smallest elements being the most precise ones. Function $\beta : \Sigma \rightarrow \Sigma^\alpha$ is the abstraction function that associates each state s with its “best” (wrt \leq^α) approximation $\beta(s) \in \Sigma^\alpha$.⁵ In the sequel, we always assume that M^α is a \mathcal{I}_α -abstract interpretation of M .

Given a trace $x = t_0 \xrightarrow{a_0} t_1 \xrightarrow{a_1} \dots$, we denote with $\beta(x)$ the abstract

⁵ We have not used α to denote this function because this name usually refers to the lower adjoint $\alpha : 2^\Sigma \rightarrow \Sigma^\alpha$ of a Galois connection between $(2^\Sigma, \subseteq)$ and $(\Sigma^\alpha, \leq^\alpha)$. Anyway, $\forall s \in \Sigma. \alpha(\{s\}) = \beta(s)$.

trace $\beta(t_0) \xrightarrow{a_0}_\alpha \beta(t_1) \xrightarrow{a_1}_\alpha \dots$. Note that possibly $\beta(x)$ is not an element of $\mathcal{O}(M^\alpha)$. In addition, given $x^\alpha = t_0^\alpha \xrightarrow{a_0}_\alpha t_1^\alpha \xrightarrow{a_1}_\alpha \dots$ and $y^\alpha = r_0^\alpha \xrightarrow{a_0}_\alpha r_1^\alpha \xrightarrow{a_1}_\alpha \dots$, we write $x^\alpha \leq^\alpha y^\alpha$ when $\forall i \geq 0. t_i^\alpha \leq^\alpha r_i^\alpha$.

Definition 2.1 We say that M^α is \mathcal{I}_α -correct wrt M , iff $\forall x \in \mathcal{O}(M)$ there exists $x^\alpha \in \mathcal{O}(M^\alpha)$ such that $\beta(x) \leq^\alpha x^\alpha$.

As usual in abstract interpretation, \mathcal{I}_α -correctness means that the abstract transition system over-approximates the original one, which may introduce *imprecision* when analyzing temporal properties over the abstract model. Also, the construction of abstract transition systems may additionally produce so-called “*spurious*” abstract traces, that is, traces that do not correspond to any concrete ones. These traces are undesirable because they can lead to obtaining false answers when analyzing temporal properties. The notion of \mathcal{I}_α -completeness defines the abstract transition systems without “*spurious*” traces.

Definition 2.2 We say that M^α is \mathcal{I}_α -complete wrt M iff $\forall x^\alpha \in \mathcal{O}(M^\alpha)$ there exists $x \in \mathcal{O}(M)$ such that $\beta(x) \leq^\alpha x^\alpha$.

Example 2.3 This example has been extracted from [5] that is one of the key references of the classic method. Consider a system composed of two processes (the *dining mathematicians*) which use a parallel version of the Colatz program for the mutually exclusive access to the critical section where they may eat. Consider the LTS $M = (\Sigma, A, \xrightarrow{\quad}, s_0)$ where the set of system states Σ is $\{\text{think}, \text{eat}\}^2 \times N$, N being the set of natural numbers. An element $\langle l_0, l_1, n \rangle \in \Sigma$ represents the state of each mathematician, *thinking* or *eating*, and the current value of variable n . The set of actions is $A = \{\text{odd}(n), \text{even}(n), \text{mult}(n), \text{div}(n)\}$, the initial state s_0 is $\langle \text{think}, \text{think}, 100 \rangle$, and the transition relation is defined as follows:

$\langle \text{think}, l_1, n \rangle \xrightarrow{\text{odd}(n)} \langle \text{eat}, l_1, n \rangle$	$\langle l_0, \text{think}, n \rangle \xrightarrow{\text{even}(n)} \langle l_0, \text{eat}, n \rangle$
$\langle \text{eat}, l_1, n \rangle \xrightarrow{\text{mult}(n)} \langle \text{think}, l_1, 3 * n + 1 \rangle$	$\langle l_0, \text{eat}, n \rangle \xrightarrow{\text{div}(n)} \langle l_0, \text{think}, n/2 \rangle$

That is, the parity of n decides which mathematician may eat.⁶

Consider the abstract LTS $M^\alpha = (\Sigma^\alpha, A, \xrightarrow{\quad}_\alpha, s_0^\alpha)$ where $\Sigma^\alpha = \{\text{think}, \text{eat}\}^2 \times \{\perp, e, o, \top\}$. The set $\{\perp, e, o, \top\}$ is a lattice with the partial order \leq^α defined as: $s_1^\alpha \leq^\alpha s_2^\alpha \Leftrightarrow s_1^\alpha = \perp$ or $s_2^\alpha = \top$. The initial state is $s_0^\alpha = \langle \text{think}, \text{think}, e \rangle$ and the abstract transition relation $\xrightarrow{\quad}_\alpha$ is given by:

⁶ We could complicate this model considering more initial states (which could be added with additional transition rules.) However, this concrete model is sufficient to illustrate the main results of the paper.

$\langle think, l_1, o \rangle \xrightarrow{\text{odd}(n)}_{\alpha} \langle eat, l_1, o \rangle$	$\langle l_0, think, e \rangle \xrightarrow{\text{even}(n)}_{\alpha} \langle l_0, eat, e \rangle$
$\langle think, l_1, \top \rangle \xrightarrow{\text{odd}(n)}_{\alpha} \langle eat, l_1, \top \rangle$	$\langle l_0, think, \top \rangle \xrightarrow{\text{even}(n)}_{\alpha} \langle l_0, eat, \top \rangle$
$\langle eat, l_1, o \rangle \xrightarrow{\text{mult}(n)}_{\alpha} \langle think, l_1, e \rangle$	$\langle eat, l_1, \top \rangle \xrightarrow{\text{mult}(n)}_{\alpha} \langle think, l_1, \top \rangle$
$\langle l_0, eat, e \rangle \xrightarrow{\text{div}(n)}_{\alpha} \langle l_0, think, \top \rangle$	$\langle l_0, eat, \top \rangle \xrightarrow{\text{div}(n)}_{\alpha} \langle l_0, think, \top \rangle$

Note that the action $div(n)$ ($n:=n/2$) produces the imprecise value \top for n . In addition, once this action has been executed, it is not possible to generate a more precise value for n .

Define $\beta : \Sigma \rightarrow \Sigma^{\alpha}$ as $\beta(\langle l_0, l_1, n \rangle) = \langle l_0, l_1, e \rangle$ iff n is even and $\beta(\langle l_0, l_1, n \rangle) = \langle l_0, l_1, o \rangle$, otherwise, and consider $\mathcal{I}_{\alpha} = (\Sigma, (\Sigma^{\alpha}, \leq^{\alpha}), \beta)$. Then M^{α} is \mathcal{I}_{α} -correct wrt M .

The unique trace in $\mathcal{O}(M)$ is

$$\begin{aligned}
 x = & \langle think, think, 100 \rangle \xrightarrow{\text{even}(n)} \langle think, eat, 100 \rangle \xrightarrow{\text{div}(n)} \\
 & \langle think, think, 50 \rangle \xrightarrow{\text{even}(n)} \langle think, eat, 50 \rangle \xrightarrow{\text{div}(n)} \\
 & \langle think, think, 25 \rangle \xrightarrow{\text{odd}(n)} \langle eat, think, 25 \rangle \dots
 \end{aligned}$$

which is abstracted by $x^{\alpha} \in \mathcal{O}(M^{\alpha})$ (that is, $\beta(x) \leq^{\alpha} x^{\alpha}$) where

$$\begin{aligned}
 x^{\alpha} = & \langle think, think, e \rangle \xrightarrow{\text{even}(n)}_{\alpha} \langle think, eat, e \rangle \xrightarrow{\text{div}(n)}_{\alpha} \\
 & \langle think, think, \top \rangle \xrightarrow{\text{even}(n)}_{\alpha} \langle think, eat, \top \rangle \xrightarrow{\text{div}(n)}_{\alpha} \\
 & \langle think, think, \top \rangle \xrightarrow{\text{odd}(n)}_{\alpha} \langle eat, think, \top \rangle \dots
 \end{aligned}$$

Trace x^{α} is imprecise since from the third state the parity of variable n has been lost, that is, $\beta(x) \neq x^{\alpha}$. $\mathcal{O}(M^{\alpha})$ also contains some “spurious” traces like

$$\begin{aligned}
 & \langle think, think, e \rangle \xrightarrow{\text{even}(n)}_{\alpha} \langle think, eat, e \rangle \xrightarrow{\text{div}(n)}_{\alpha} \\
 & \langle think, think, \top \rangle \xrightarrow{\text{odd}(n)}_{\alpha} \langle eat, think, \top \rangle \xrightarrow{\text{mult}(n)}_{\alpha} \\
 & \langle think, think, \top \rangle \xrightarrow{\text{even}(n)}_{\alpha} \langle think, eat, \top \rangle \dots
 \end{aligned}$$

Now consider $M_p^{\alpha} = (\Sigma^{\alpha}, A, \xrightarrow{\text{p}}_{\alpha}, s_0^{\alpha})$, where the relation $\xrightarrow{\text{p}}_{\alpha}$ is:

$\langle think, l_1, o \rangle \xrightarrow{\text{odd}(n)^{\text{p}}}_{\alpha} \langle eat, l_1, o \rangle$	$\langle l_0, think, e \rangle \xrightarrow{\text{even}(n)^{\text{p}}}_{\alpha} \langle l_0, eat, e \rangle$
$\langle eat, l_1, o \rangle \xrightarrow{\text{mult}(n)^{\text{p}}}_{\alpha} \langle think, l_1, e \rangle$	
$\langle l_0, eat, e \rangle \xrightarrow{\text{div}(n)^{\text{p}}}_{\alpha} \langle l_0, think, e \rangle$	$\langle l_0, eat, e \rangle \xrightarrow{\text{div}(n)^{\text{p}}}_{\alpha} \langle l_0, think, o \rangle$

Observe that in this example the imprecision of the action $div(n)$ is solved by means of a non-deterministic selection between the two rules written in bold. Thus, $\mathcal{O}(M_p^\alpha)$ contains more abstract traces than $\mathcal{O}(M^\alpha)$, but in contrast, the traces in $\mathcal{O}(M_p^\alpha)$ are more precise. M_p^α is also \mathcal{I}_α -correct wrt M , and the abstract trace approximating $x \in \mathcal{O}(M)$ is

$$\begin{aligned} x_1^\alpha &= \langle think, think, e \rangle \xrightarrow{\alpha}^{even(n)^p} \langle think, eat, e \rangle \xrightarrow{\alpha}^{div(n)^p} \\ &\quad \langle think, think, e \rangle \xrightarrow{\alpha}^{even(n)^p} \langle think, eat, e \rangle \xrightarrow{\alpha}^{div(n)^p} \\ &\quad \langle think, think, o \rangle \xrightarrow{\alpha}^{odd(n)^p} \langle eat, think, o \rangle \dots \end{aligned}$$

Note that $\beta(x) = x_1^\alpha$. Furthermore, M_p^α also contains “spurious” traces. In contrast, a transition system M_1^α such that $\mathcal{O}(M_1^\alpha) = \{x^\alpha, x_1^\alpha\}$ would be \mathcal{I}_α -complete wrt M . However note that, for this example, it is not easy to define an abstract transition system generating a complete model.

3 Abstracting Temporal Logic

Kripke structures are used to evaluate temporal formulas against models. In this section, we summarize the classic approach for abstracting Kripke structures and also discuss the main preservation results that may be deduced from this definition. In order to easily integrate the classic method and our proposal, we consider *weak* Kripke structures where the *negation* \neg is not dealt with as a connective, but as a way of constructing an atomic proposition.

3.1 Temporal Logic

Given $Prop$ a set of propositions, we construct the set $\mathcal{P} = Prop \cup \neg Prop$, where $\neg Prop = \{\neg p : p \in Prop\}$. Let \mathcal{F} be the set of LTL temporal formulas built inductively using the elements of \mathcal{P} , the standard Boolean operators, except \neg , and the *temporal operators*: *next* “ \bigcirc ”, *always* “ \square ”, *eventually* “ \diamond ” and *until* “ U ”.

A LTS $M = (A, \Sigma, \xrightarrow{\quad}, s_0)$ may be extended to a *weak Kripke* structure $\mathcal{K} = \langle M, \tau \rangle$ where $\tau : \Sigma \rightarrow 2^{\mathcal{P}}$ is a function that assigns truth values to the propositions of \mathcal{P} in each state.

$\mathcal{K} = \langle M, \tau \rangle$ is a Kripke structure iff $\forall s \in \Sigma, \forall p \in Prop$ the *Principle of Excluded Middle* (PEM) (i. e., $p \in \tau(s) \vee \neg p \in \tau(s)$), and the *Principle of Non-Contradiction* (PNC) (i. e., $p \notin \tau(s) \vee \neg p \notin \tau(s)$) hold.

Note that \mathcal{K} defines an interpretation of actions and an interpretation of atomic propositions. In the following, $p \in \mathcal{P}$ denotes both non-negated and negated atomic propositions.

Definition 3.1 Let $\mathcal{K} = \langle M, \tau \rangle$ be a weak Kripke/Kripke structure. Given a trace $x = t_0 \xrightarrow{a_0} \dots$, and properties $p \in \mathcal{P}$ and $f, g \in \mathcal{F}$, we define relation

\models^τ inductively as follows:

$$\begin{aligned}
 x \models^\tau p & \quad \text{iff } p \in \tau(t_0). \\
 x \models^\tau f \vee g & \quad \text{iff } x \models^\tau f \text{ or } x \models^\tau g. \\
 x \models^\tau f \wedge g & \quad \text{iff } x \models^\tau f \text{ and } x \models^\tau g. \\
 x \models^\tau f \rightarrow g & \quad \text{iff } x \models^\tau f \text{ implies } x \models^\tau g. \\
 x \models^\tau \bigcirc f & \quad \text{iff } x^1 \models^\tau f. \\
 x \models^\tau \square f & \quad \text{iff } \forall k \geq 0. x^k \models^\tau f. \\
 x \models^\tau \diamond f & \quad \text{iff } \exists k \geq 0. x^k \models^\tau f. \\
 x \models^\tau f U g & \quad \text{iff } \exists k \geq 0. (x^k \models^\tau g \text{ and } \forall j < k. [x^j \models^\tau f]).
 \end{aligned}$$

Finally, we extend \models^τ to weak Kripke structures as follows.

- (i) *Universal formulas:* $M \models^\tau \forall f$ iff $\forall x \in \mathcal{O}(M). x \models^\tau f$.
- (ii) *Existential formulas:* $M \models^\tau \exists f$ iff $\exists x \in \mathcal{O}(M). x \models^\tau f$.

In the following sections, we always assume that the abstract LTS $M^\alpha = (A, \Sigma^\alpha, \xrightarrow{\alpha}, s_0^\alpha)$ is \mathcal{I}_α -correct wrt $M = (A, \Sigma, \xrightarrow{\quad}, s_0)$. In addition, in order to simplify notation, we will write \models and \models^α instead of \models^τ and \models^{τ^α} , respectively. Besides, to differentiate between the two different ways of abstracting the satisfiability relation considered in the paper, we will use \models_c^α when we refer to the classic relation.

3.2 The classic method

Let $\mathcal{K} = \langle M, \tau \rangle$ and $\mathcal{K}^\alpha = \langle M^\alpha, \tau_c^\alpha \rangle$ be two weak Kripke structures. The classic way of defining $\tau_c^\alpha(s^\alpha)$ is as follows⁷

$$\tau_c^\alpha(s^\alpha) = \bigcap \{ \tau(s) \mid \beta(s) \leq^\alpha s^\alpha \} \quad (\text{Under}_c)$$

Usually, \mathcal{K} is a Kripke structure, that is, it satisfies the conditions PNC and PEM. However, note that the way of defining τ_c^α makes it possible that for a given abstract state s^α and a proposition $p \in \mathcal{P}$, neither $p \in \tau_c^\alpha(s^\alpha)$ nor $\neg p \in \tau_c^\alpha(s^\alpha)$ occur.

Definition (Under_c) has some interesting properties, such as,

- (i) τ_c^α under-approximates τ ,

$$\beta(s) \leq^\alpha s^\alpha \Rightarrow \tau(s) \supseteq \tau_c^\alpha(s^\alpha) \quad (LC_c)$$

- (ii) τ_c^α is the biggest set verifying the condition (LC_c) ,

$$\text{if } \exists \mathcal{Q} \subseteq \mathcal{P}. \forall s (\beta(s) \leq^\alpha s^\alpha \Rightarrow \tau(s) \supseteq \mathcal{Q}) \text{ then } \tau_c^\alpha(s^\alpha) \supseteq \mathcal{Q} \quad (C_c)$$

⁷ Note that the codomain of both τ and τ_c^α coincide.

(iii) τ_c^α is monotonic decreasing,

$$\text{if } s_1^\alpha \leq^\alpha s_2^\alpha \text{ and } \exists s. \beta(s) \leq^\alpha s_1^\alpha \text{ then } \tau_c^\alpha(s_1^\alpha) \supseteq \tau_c^\alpha(s_2^\alpha) \quad (M_c)$$

(iv) LC_c and C_c univocally determine τ_c^α ,

$$\text{Under}_c \iff LC_c \text{ and } C_c \quad (E_c)$$

(v) The extension to abstract traces preserves the satisfiability relation from the abstract to the concrete model, that is, given $f \in \mathcal{F}$

$$\beta(x) \leq^\alpha x^\alpha \Rightarrow (x^\alpha \models_c^\alpha f \Rightarrow x \models f) \quad (Cons_c)$$

Proof. [of properties of Definition Under_c]

(i) Conditions LC_c , C_c and E_c hold by definition of τ_c^α .

(iii) Condition M_c : If $s_1^\alpha \leq^\alpha s_2^\alpha$, then $\{s | \beta(s) \leq^\alpha s_1^\alpha\} \subseteq \{s | \beta(s) \leq^\alpha s_2^\alpha\}$. Thus, assuming that $\{s | \beta(s) \leq^\alpha s_1^\alpha\} \neq \emptyset$, we have that $\bigcap \{\tau(s) | \beta(s) \leq^\alpha s_1^\alpha\} \supseteq \bigcap \{\tau(s) | \beta(s) \leq^\alpha s_2^\alpha\}$, or equivalently, $\tau_c^\alpha(s_1^\alpha) \supseteq \tau_c^\alpha(s_2^\alpha)$.

(v) Condition $Cons_c$: Consider that $f = p$ is an atomic proposition. Let us assume $\beta(x) \leq^\alpha x^\alpha$ where $x^\alpha = t_0^\alpha \rightarrow \dots$ and $x = t_0 \rightarrow \dots$, then $\beta(t_0) \leq^\alpha t_0^\alpha$, which implies by LC_c that $\tau_c^\alpha(t_0^\alpha) \subseteq \tau(t_0)$. By definition, if $x^\alpha \models_c^\alpha p$, then $p \in \tau_c^\alpha(t_0^\alpha)$, that is, $p \in \tau(t_0)$. Finally, applying the definition of \models , we obtain $x \models p$. The rest of the cases are proved by induction on the formula structure. \square

Condition $Cons_c$ assures the weak conservation of universal properties from the abstract to the concrete model:

Theorem 3.2 *Given $f \in \mathcal{F}$, if $M^\alpha \models_c^\alpha \forall f$ then $M \models \forall f$.*

Proof. Let $x \in \mathcal{O}(M)$. By the \mathcal{I}_α -correctness condition, there exists $x^\alpha \in \mathcal{O}(M^\alpha)$ such that $\beta(x) \leq^\alpha x^\alpha$. Since by hypothesis $M^\alpha \models_c^\alpha \forall f$, then we have that $x^\alpha \models_c^\alpha f$. Finally, using Condition $Cons_c$, we deduce $x \models f$. \square

Example 3.3 Following Example 2.3, consider propositions $even(n), odd(n) \in \mathcal{P}$. If we denote with $s^\alpha.n$ the value of variable n in the abstract state s^α then, using Definition Under_c , we have that $odd(n) \in \tau_c^\alpha(s^\alpha) \iff s^\alpha.n = o$, and $even(n) \in \tau_c^\alpha(s^\alpha) \iff s^\alpha.n = e$. In addition, by Theorem 3.2, checking $M_p^\alpha \models_c^\alpha \forall \diamond odd(n)$ implies that $M \models \forall \diamond odd(n)$. This property means that the first mathematician is not delayed indefinitely. Note that with a real model, which generates more traces, this result is very interesting.

3.3 The Over-Approximation Method

Consider now the dual abstraction of \mathcal{K} given by $\mathcal{K}^\alpha = \langle M^\alpha, \tau^\alpha \rangle$ where

$$\tau^\alpha(s^\alpha) = \bigcup \{ \tau(s) | \beta(s) \leq^\alpha s^\alpha \} \quad (Over)$$

Note that with this definition, it is possible that for a given abstract state s^α and a proposition $p \in \mathcal{P}$, either $p \in \tau^\alpha(s^\alpha)$ and $\neg p \in \tau^\alpha(s^\alpha)$ occur. The following properties of τ^α are dual to the ones given in the preceding section:

- (i) τ^α over-approximates τ ,

$$\beta(s) \leq^\alpha s^\alpha \Rightarrow \tau(s) \subseteq \tau^\alpha(s^\alpha) \quad (LC)$$

- (ii) τ^α is the smallest set verifying the condition (LC),

$$\text{if } \exists \mathcal{Q} \subseteq \mathcal{P}. \forall s (\beta(s) \leq^\alpha s^\alpha \Rightarrow \tau(s) \subseteq \mathcal{Q}) \text{ then } \tau^\alpha(s^\alpha) \subseteq \mathcal{Q} \quad (C)$$

- (iii) τ^α is monotonic increasing,

$$\text{if } s_1^\alpha \leq^\alpha s_2^\alpha \text{ then } \tau^\alpha(s_1^\alpha) \subseteq \tau^\alpha(s_2^\alpha) \quad (M)$$

- (iv) LC and C univocally determine τ^α ,

$$\text{Over} \iff LC \text{ and } C \quad (E)$$

- (v) The extension to abstract traces preserves the satisfiability relation from the concrete to the abstract model, that is, given $f \in \mathcal{F}$

$$\beta(x) \leq^\alpha x^\alpha \Rightarrow (x \models f \Rightarrow x^\alpha \models^\alpha f) \quad (Cons)$$

Proof. [of properties of Definition *Over*]

- (i) Conditions LC, C and E hold by definition of τ^α .

- (iii) Condition M: If $s_1^\alpha \leq^\alpha s_2^\alpha$ then $\{s | \beta(s) \leq^\alpha s_1^\alpha\} \subseteq \{s | \beta(s) \leq^\alpha s_2^\alpha\}$. Thus, we have that $\bigcup \{\tau(s) | \beta(s) \leq^\alpha s_1^\alpha\} \subseteq \bigcup \{\tau(s) | \beta(s) \leq^\alpha s_2^\alpha\}$, or equivalently, $\tau^\alpha(s_1^\alpha) \subseteq \tau^\alpha(s_2^\alpha)$.

- (v) Condition *Cons*: Consider that $f = p$ is an atomic proposition. Let us assume $\beta(x) \leq^\alpha x^\alpha$ where $x^\alpha = t_0^\alpha \rightarrow \dots$ and $x = t_0 \rightarrow \dots$, then $\beta(t_0) \leq^\alpha t_0^\alpha$, which implies by LC that $\tau(t_0) \subseteq \tau^\alpha(t_0^\alpha)$. By definition, if $x \models p$ then $p \in \tau(t_0)$, that is, $p \in \tau^\alpha(t_0^\alpha)$. Finally, by definition, we obtain $x^\alpha \models^\alpha p$. The rest of the cases are proved by induction on the formula structure. □

Condition *Cons* assures the weak refutation of existential properties from the abstract to the concrete model:

Theorem 3.4 *Given $f \in \mathcal{F}$, if $M^\alpha \not\models^\alpha \exists f$ then $M \not\models \exists f$.*

Proof. Let $x \in \mathcal{O}(M)$. By the \mathcal{I}_α -correctness condition, there exists $x^\alpha \in \mathcal{O}(M^\alpha)$ such that $\beta(x) \leq^\alpha x^\alpha$. Assume that $x \models f$, then applying condition *Cons*, we obtain that $x^\alpha \models^\alpha f$. But this is not possible by hypothesis. Therefore, $x \not\models f$. □

Example 3.5 Following Examples 2.3 and 3.3, by Definition *Over*, we have that for the over-approximation method $odd(n) \notin \tau^\alpha(s^\alpha) \iff s^\alpha.n = e$ and $even(n) \notin \tau^\alpha(s^\alpha) \iff s^\alpha.n = o$. We exclude the abstract value \perp because it is never reached by the abstract traces. Thus, by Theorem 3.4, proving that

$M_p^\alpha \not\models^\alpha \exists \square \text{even}(n)$ implies that $M \not\models \exists \square \text{even}(n)$. Note that both approaches prove the same property but using dual methods. The selection of the method depends on the property to be analyzed. If formula f represents a desired property that should be held by all traces, then you must use the classic method. On the contrary, if f represents an erroneous behavior that no trace should satisfy then you must use the over-approximation method.

For instance, assume that you want to check the following property over the model M : “It never occurs that variable n is odd and the second mathematician is eating”. You may specify this property as an erroneous behaviour with the temporal formula: $f = \diamond(\text{odd}(n) \wedge l_1 = \text{eat})$. Now, checking $M_p^\alpha \not\models^\alpha \exists \diamond(\text{odd}(n) \wedge l_1 = \text{eat})$ proves that no concrete trace matches this erroneous behaviour. Alternatively, you may specify the desired behaviour as $f = \square(\text{even}(n) \vee l_1 = \text{think})$ and check $M_p^\alpha \models_c^\alpha \forall \square(\text{even}(n) \vee l_1 = \text{think})$ to prove that all traces have the expected behaviour. Note that for this example it seems easier to specify the “bad state” than the “good one”.

Note that Theorems 3.2 and 3.4 are not equivalent because both methods deal with negation using non-standard and dual approaches. Thus, considering that formula $\neg f$ is in negation normal form, we have that $M^\alpha \not\models_c^\alpha \forall f \not\Rightarrow M^\alpha \models_c^\alpha \exists \neg f$, and, in addition, $M^\alpha \models^\alpha \forall f \not\Rightarrow M^\alpha \not\models^\alpha \exists \neg f$.

4 Incompleteness and Imprecision

Assuming completeness, we achieve dual results for the classic and over-approximation methods concerning the satisfaction (resp. refutation) of existential (resp. universal) properties.

Theorem 4.1 *If M^α is \mathcal{I}_α -complete wrt M then given $f \in \mathcal{F}$,*

- (1) $M^\alpha \models_c^\alpha \exists f \Rightarrow M \models \exists f$
- (2) $M^\alpha \not\models^\alpha \forall f \Rightarrow M \not\models \forall f$

Proof.

- (i) By hypothesis, there exists $x^\alpha \in \mathcal{O}(M^\alpha)$ such that $x^\alpha \models_c^\alpha f$. As M^α is \mathcal{I}_α -complete wrt M then there exists $x \in \mathcal{O}(M)$ such that $\beta(x) \leq^\alpha x^\alpha$. Finally, by condition $Cons_c$, since $x^\alpha \models_c^\alpha f$ we have that $x \models f$, that is, $M \models \exists f$.
- (ii) By hypothesis, there exists $x^\alpha \in \mathcal{O}(M^\alpha)$ such that $x^\alpha \not\models^\alpha f$. As M^α is \mathcal{I}_α -complete wrt M then there exists $x \in \mathcal{O}(M)$ such that $\beta(x) \leq^\alpha x^\alpha$. Finally, by condition $Cons$, since $x^\alpha \not\models^\alpha f$ we deduce that $x \not\models f$, that is, $M \not\models \forall f$.

□

The previous theorem is mainly used for debugging, since if the model checker provides an abstract trace, we know that this is not spurious.

We now discuss how the imprecision and incompleteness with which the abstract model has been defined affects the analysis of properties. Imprecision occurs when an abstract trace x^α approximating a concrete one x is strictly bigger (wrt \leq^α) than $\beta(x)$. Considering the classic method, due to condition M_c , bigger states may lose information about the satisfiability of propositions. In addition, in the over-approximation method, condition M means the same but in the opposite direction. Thus we have

$$(a) M^\alpha \not\models_c^\alpha \forall f \not\Rightarrow M \not\models \forall f \text{ vs. } M^\alpha \models^\alpha \forall f \not\Rightarrow M \models \forall f$$

$$(b) M^\alpha \not\models_c^\alpha \exists f \not\Rightarrow M \not\models \exists f \text{ vs. } M^\alpha \models^\alpha \exists f \not\Rightarrow M \models \exists f$$

Example 4.2 Consider the imprecise abstract model M^α defined in Example 2.3 the property $f = \diamond odd(n)$ that express that variable n eventually takes an odd value. Considering the classic method, due to the imprecision of the model, there exist some abstract traces for which this property does not hold, that is, $M^\alpha \not\models_c^\alpha \forall \diamond odd(n)$. For instance, the trace x^α defined in Example 2.3, does not satisfy f , since $odd(n) \notin \tau_c^\alpha(s^\alpha)$ when $s^\alpha.n = \top$. However note that $M \models \forall \diamond odd(n)$ as studied in Example 3.3. Alternatively consider the over-approximation method and the same abstract model M^α . For the abstract trace x^α , we have that $x^\alpha \models^\alpha \square even(n)$ since $even(n) \in \tau^\alpha(s^\alpha)$, when $s^\alpha.n = \top$. Therefore $M^\alpha \models^\alpha \exists \square even(n)$. However, $M \not\models \exists \square even(n)$ as studied in Example 3.5.

In the rest of the section, we prove that when the abstract model is *precise wrt the property to be analyzed*, both methods produce equivalent results. As a previous result, we study the effect of the abstract interpretation over the meaning of the properties to be checked.

Definition 4.3 [Strong Consistency Condition] τ^α is strongly consistent wrt τ and α when the following condition holds:

$$\tau(s) \subseteq \tau^\alpha(s^\alpha) \Rightarrow \beta(s) \leq^\alpha s^\alpha \quad (SC)$$

Note that *consistency* condition LC and SC are reciprocal conditions.

Definition 4.4 [Abstract Implication] Given $f_1, f_2 \in \mathcal{F}$ then $f_1 \Rightarrow_\alpha f_2$ iff $\forall x^\alpha \in \mathcal{O}(M^\alpha) : x^\alpha \models^\alpha f_1 \Rightarrow x^\alpha \models^\alpha f_2$.

Proposition 4.5 Let $\mathcal{K} = \langle Std, \tau \rangle$ and $\mathcal{K}^\alpha = \langle Std^\alpha, \tau^\alpha \rangle$ be two weak Kripke structures such that \mathcal{K}^α is \mathcal{I}_α -correct wrt \mathcal{K} , τ^α being strongly consistent wrt τ and \mathcal{I}_α . Then, given $x \in \mathcal{O}(M)$, and $f \in \mathcal{F}$ such that $\beta(x) \models^\alpha f$, there exists a formula $f' \in \mathcal{F}$ such that $x \models f'$ and $f' \Rightarrow_\alpha f$.

Proof. In [8]. □

Consider an atomic proposition $p \in \mathcal{P}$ to be analyzed against the abstract model M^α . If we use the classic method, we need impose that if

$p \in \tau(s)$ then $p \in \tau_c^\alpha(\beta(s))$. Otherwise, we would have that the information about p has been lost even before of the analysis, and therefore, this is not useful. For instance, the abstract interpretation \mathcal{I}_α defined in Example 2.3 is not adequate for analyzing property $f = \diamond(n > 10)$, since the abstraction has lost all the information about $n > 10$ from the beginning. This is also applicable to the over-approximation method. However Proposition 4.5 states that when condition SC holds, this method may use some properties that seem to be inconsistent with the abstraction. For instance, consider the formula $\square(n = 2)$. Clearly, the classic method cannot analyze it. However, for the dual method the situation is different. Note that $n = 2 \notin \tau^\alpha(s^\alpha) \iff s^\alpha.n = o \iff \text{even}(n) \notin \tau^\alpha(s^\alpha)$. Thus, checking $M^\alpha \not\models^\alpha \exists \square(n = 2)$ is equivalent to prove that $M^\alpha \not\models^\alpha \exists \square \text{even}(n)$.

The previous proposition allows us to define the notion of abstract extension of a formula f .

Definition 4.6 [Abstract extension of a formula] Given $f \in \mathcal{F}$, we define \overline{f}^α , the *abstract extension* of f , as $\bigvee \{f' \mid f' \Rightarrow_\alpha f\}$.

For instance, $\overline{n = 2}^\alpha = \bigvee \{n = a \mid \text{even}(a)\} = \text{even}(n)$. However, in general, there is no guarantee that formula \overline{f}^α can be constructed. We only use the operator $\overline{\cdot}^\alpha$ to represent the set of concretizations of a given formula. It denotes the possible loss of precision of f due to the abstract interpretation, that is $f \Rightarrow \overline{f}^\alpha$, which can be easily proved. In general, the opposite is not true, and f and \overline{f}^α do not coincide; this means that \mathcal{I}_α has modified the meaning of f , which cannot be restored.

The next definition captures the loss of information due to the existence of imprecise abstract traces in the abstract model.

Definition 4.7 We say that formula $f \in \mathcal{F}$ does not *lose precision* wrt $\mathcal{O}(M^\alpha)$ iff $\forall x^\alpha \in \mathcal{O}(M^\alpha)$, if $x^\alpha \models^\alpha f$ then $x^\alpha \not\models^\alpha \neg f$.

Proposition 4.8 Assume that \mathcal{K} is a Kripke structure⁸ and that formula $f \in \mathcal{F}$ does not lose precision wrt $\mathcal{O}(M^\alpha)$, then

$$M^\alpha \models^\alpha \forall f \Rightarrow M \models \forall \overline{f}^\alpha$$

Proof. Assume that $f = p \in \mathcal{P}$. Given $x \in \mathcal{O}(M)$, by the \mathcal{I}_α -correctness, there exists $x^\alpha \in \mathcal{O}(M^\alpha)$ such that $\beta(x) \leq^\alpha x^\alpha$. It is easy to prove that $\beta(x) \models^\alpha \neg p$ implies that $x^\alpha \models^\alpha \neg p$. But this is not possible by Definition 4.7, thus $\beta(x) \not\models^\alpha \neg p$. Since, \mathcal{K} is a Kripke structure, this implies that $\beta(x) \models p$. Finally, by Proposition 4.5, we have that $x \models \overline{p}^\alpha$. The rest of the cases are proved by induction on the formula structure. \square

The previous proposition states that when the formula f to be analyzed has not lost precision wrt the abstract model, the over approximation method may be used to prove the universal formula $\forall f$. We can automatically check

⁸ This condition may be weakened but we use it here to simplify the proof

if a given formula loses precision wrt a given abstract model. Although the development of this result is out of the scope of the present paper, the method is based on checking the abstract values that taken by the variables during the model checking process.

Finally, assuming that the formula f to be checked has not lost information due to the abstract interpretation, that is, $f = \overline{f}^\alpha$, and that the abstract model has not lost information wrt f we have that classic and over-approximation methods coincide when analyzing the satisfaction of universal formulas, that is, $M^\alpha \models_c^\alpha \forall f \Leftrightarrow M^\alpha \models^\alpha \forall f$.

Example 4.9 In the context of the previous examples, consider the formula $f = \Box(l_0 = think \vee l_1 = think)$. Since abstraction does not modify f , we have that f does not loss precision wrt model M_p^α , and also that $\overline{f}^\alpha = f$. In these conditions, proving $M_p^\alpha \models^\alpha \forall \Box(l_0 = think \vee l_1 = think)$ is equivalent to proving $M_p^\alpha \models_c^\alpha \forall \Box(l_0 = think \vee l_1 = think)$.

5 Discussion and Conclusion

From our experience in the verification of concurrent systems using temporal logic, we have noted that many properties are naturally expressed as universal formulas, but others are more easily written to be refuted. This observation agrees with the functionality implemented in many model checking tools like SPIN [9]. This leads us to argue that the ideal method for abstracting properties when realizing abstract model checking should integrate the classic and our over-approximation method. The classic method could increase the confidence in the quality of software by proving the satisfaction of some key properties, and our method could be used to discard very critical errors.

When analyzing universal properties, we may consider that the classical and the over-approximation methods produce the most and the least precise abstract satisfiability relations, respectively. To this respect, we have developed an approach that allows users to obtain intermediate precision results when these two methods do not provide definite answers, avoiding the construction of a new abstract model. We are currently extending our tool α SPIN [6,7] <http://www.lcc.uma.es/~gisum/fmse/tools> to incorporate this capability.

Acknowledgements We would like to thank the reviewers for their valuable comments to improve this paper.

References

- [1] E. Clarke, O. Grumberg, D. Peled, *Model Checking* The MIT Press, 2000. 1
- [2] P. Cousot, R. Cousot, Abstract Interpretation: A Unified Lattice Model for Static Analysis of Programs by Construction or Approximation of Fixpoints, in

- Conf. Record of the 4th ACM Principles of Programming Languages conference*, 1977, pp. 238–252. [1](#)
- [3] E.M. Clarke, O. Grumberg, D.E. Long, Model Checking and Abstraction, *ACM Transactions on Programming Languages and Systems (TOPLAS)*, **16**(5) (1994) 1512–1245. [1](#)
- [4] E.M. Clarke, O. Grumberg, S. Jha, Y. Lu, and H. Veith. Counterexample-guided abstraction refinement. In *Proc. of 12th International Conference on Computer Aided Verification 2000*, pp. 154–169. [1](#)
- [5] D. Dams, R. Gerth, O. Grumberg, Abstract Interpretation of Reactive Systems, *ACM Transactions on Programming Languages and Systems (TOPLAS)*, **19**(2) (1997) 253–291. [1](#), [2.3](#)
- [6] M.M. Gallardo, P. Merino, A Framework for Automatic Construction of Abstract PROMELA Models, *Theoretical and Practical Aspects of SPIN Model Checking*, LNCS-1680, 1999, pp. 184–199. [5](#)
- [7] M.M. Gallardo, J.Martinez, P.Merino, E.Rosales, Using XML to implement Abstraction for Model Checking. In *Proc. of the ACM Symposium on Applied Computing 2002*, pp. 1021–1025. [5](#)
- [8] M.M. Gallardo, P. Merino, E. Pimentel, Verifying Abstract LTL Properties on Concurrent Systems *Proc. of the 6th World Conference on Integrated Design & Process Technology 2002*. [1](#), [4](#)
- [9] G.J. Holzmann, The Model Checker SPIN, *IEEE Transactions on Software Engineering* **23**(5) (1997) 279–295. [5](#)
- [10] C. Loiseaux , S. Graf, J. Sifakis, A. Boujjani, S. Bensalem, Property Preserving Abstractions for the Verification of Concurrent Systems. *Formal Methods in System Design* **6** (1995) 1–35. [1](#)
- [11] C.S. Pasareanu, M.B. Dwyer and W. Visser, Finding Feasible Counter-examples when Model Checking Java Programs, In *Proc. of the 7th Tools and Algorithms for the Construction and Analysis of Systems* LNCS-2031, 2001, pp. 284-298. [1](#)
- [12] Z. Manna , A. Pnueli, *The Temporal Logic of Reactive and Concurrent Systems - Specification*, Springer-Verlag, New York, 1992. [1](#)