From model checking to a temporal proof for partial models: preliminary example

Anna Bernasconi[1], Claudio Menghi[2], Paola Spoletini[3], Lenore D. Zuck[4], and Carlo Ghezzi[1]

[1] Politecnico di Milano, DEIB - DEEPSE group,
{anna.bernasconi, carlo.ghezzi}@polimi.it
[2] Chalmers University of Technology | University of Gothenburg
claudio.menghi@gu.se
[3] Kennesaw State University,
pspoleti@kennesaw.edu
[4] University of Illinois at Chicago,
lenore@cs.uic.edu

Abstract. This paper describes in detail the example introduced in the preliminary evaluation of THRIVE. Specifically, it evaluates THRIVE over an abstraction of the ground model proposed for a critical component belonging to a medical device used by optometrists and ophtalmologits to dected visual problems.

We provide the full description of the example introduced in the preliminary evaluation of [2]. Specifically, we evaluate THRIVE over an abstraction of the ground model proposed in [1], a critical component belonging to a medical device used by optometrists and ophtalmologits to dected visual problems. In the following we describe the considered partial model, the property of interest and the deductive verification procedure performed by THRIVE over the incomplete model.

Partial model. The ground model proposed in [1] is a critical component that measures the stereoacuity of young patients. The criticality of the system resides in certifying a certain level of stereoacuity in a consistent way, such that the treatment given by the doctor to his/her patient is correct.

We provide in Figure 1 the complete Partial Kripke Structure that represents the system. In each state the propositions are indicated with their truth value. In the complete version proposed in [1] all propositions had a true/false value. Note that, in this abstracted version, in the states s_6 and s_7 the propositions related to the assessed level have an *unknown* value, meaning that the designer is currently not sure on whether the propositions should be *true* or $false$ in these states.

The propositions $\overline{fl}$ and $\overline{cert}$ that are specified on the side of each state are the complement-closed version of fl and $cert$. These propositions are used by THRIVE during the computation of the intersection of the model states with the property states.

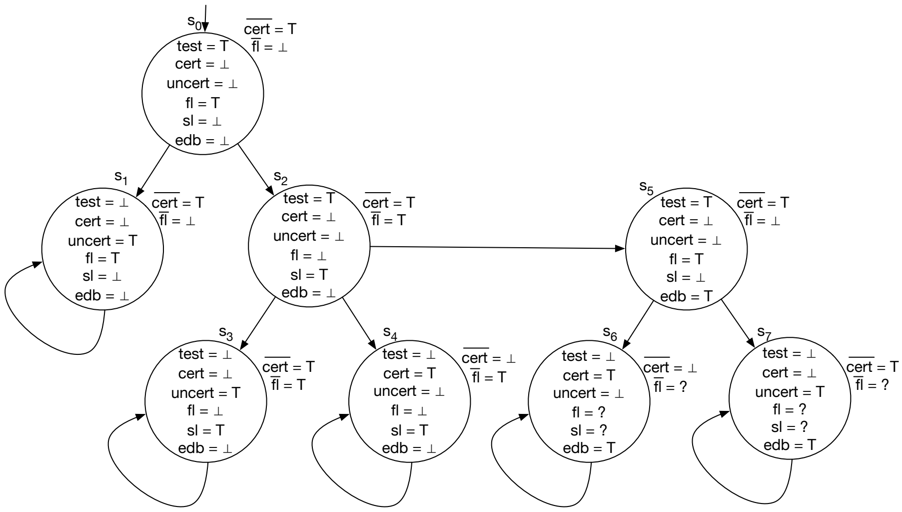

Fig. 1: Model M

Property. The property of interest is expressed by the LTL formula $\psi_3 = \Box(edb \rightarrow \Diamond(cert \vee fl))$, which states that, if an error has been made by the patient (edb) he/she cannot be uncertified and be at the second level ($\neg fl$). Indeed, a mistake prevents a patient from increasing the assessed level. Figure 2 represents the Büchi automaton corresponding to $\neg\psi_3$.

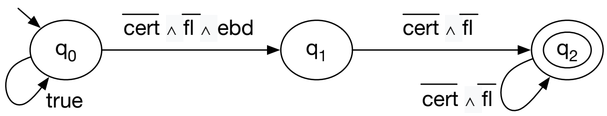

Fig. 2: The automaton $\mathcal{A}_{\neg\psi_3}$

Running THRIVE. First, the framework performs a classical model checking run on the pessimistic approximation (generated by assigning $\perp$ to the propositions fl and sl in the mentioned two states of the model). This particular assignment allows the system to reach the accepting state of the negated property q_2 in which holds $\eta(q_2) = \Box(\neg cert \wedge \neg fl)$. The returned counterexample corresponds to the path $s_0, s_2, s_5, s_7^\omega$ on the states of the model, and to the path $\langle s_0, q_0\rangle, \langle s_2, q_0\rangle, \langle s_5, q_0\rangle$, $\langle s_7, q_0\rangle, \langle s_7, q_1\rangle, \langle s_7, q_2\rangle^\omega$ on the states of the intersection space $M_{pes} \otimes \mathcal{A}_{\neg\psi_3}$. The generated accepting loop leads to conclude that $M_{pes} \not\models \psi_3$.

The framework therefore performs another model checking run on the optimistic approximation (assigning $\top$ to the unknown propositions fl and sl). This

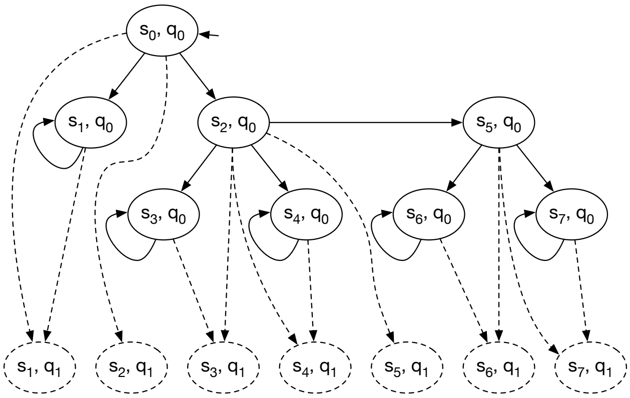

Fig. 3: The intersection automaton $M_{opt} \otimes \mathcal{A}_{\neg\psi_3}$

time the intersection state space does not contain any accepting behavior with respect to the negation of property ψ_3. The intersection space is represented in Figure 3. Table 1 presents a formal proof which shows that the optimistic approximation satisfies the property under analysis.

Table 1: Proof that ψ_3 is not violated.

Step	**Component**	**Rule**
Fail	$\langle s_1, q_1 \rangle$, $\langle s_2, q_1 \rangle$, $\langle s_3, q_1 \rangle$, $\langle s_4, q_1 \rangle$, $\langle s_5, q_1 \rangle$, $\langle s_6, q_1 \rangle$, $\langle s_7, q_1 \rangle$	$s_1 \in \mathcal{F}(I_{opt})$ $s_2 \in \mathcal{F}(I_{opt})$ $s_3 \in \mathcal{F}(I_{opt})$ $s_4 \in \mathcal{F}(I_{opt})$ $s_5 \in \mathcal{F}(I_{opt})$ $s_6 \in \mathcal{F}(I_{opt})$ $s_7 \in \mathcal{F}(I_{opt})$ ――― $s_1 \models \mu(q_1) = \neg edb \vee \bigcirc \Diamond (cert \vee fl)$ $s_2 \models \mu(q_1) = \neg edb \vee \bigcirc \Diamond (cert \vee fl)$ $s_3 \models \mu(q_1) = \neg edb \vee \bigcirc \Diamond (cert \vee fl)$ $s_4 \models \mu(q_1) = \neg edb \vee \bigcirc \Diamond (cert \vee fl)$ $s_5 \models \mu(q_1) = \neg edb \vee \bigcirc \Diamond (cert \vee fl)$ $s_6 \models \mu(q_1) = \neg edb \vee \bigcirc \Diamond (cert \vee fl)$ $s_7 \models \mu(q_1) = \neg edb \vee \bigcirc \Diamond (cert \vee fl)$
Induction	$\mathcal{X}_1 = \{\langle s_6, q_0 \rangle\}$, $Exit(\mathcal{X}_1) = \{\langle s_6, q_1 \rangle\}$	$s_6 \models \mu(q_1)$ $s_6 \rightarrow \{s_6\}$ ――― $s_6 \models \mu(q_0) = \Box(edb \rightarrow \Diamond(cert \vee fl))$
Induction	$\mathcal{X}_2 = \{\langle s_7, q_0 \rangle\}$, $Exit(\mathcal{X}_2) = \{\langle s_7, q_1 \rangle\}$	$s_7 \models \mu(q_1)$ $s_7 \rightarrow \{s_7\}$ ――― $s_7 \models \mu(q_0) = \Box(edb \rightarrow \Diamond(cert \vee fl))$

Induction	$\mathcal{X}_3 = \{\langle s_3, q_0\rangle\}$, $Exit(\mathcal{X}_3) = \{\langle s_3, q_1\rangle\}$	$\dfrac{s_3 \models \mu(q_1) \quad s_3 \to \{s_3\}}{s_3 \models \mu(q_0) = \Box(edb \to \Diamond(cert \vee fl))}$
Induction	$\mathcal{X}_4 = \{\langle s_4, q_0\rangle\}$, $Exit(\mathcal{X}_4) = \{\langle s_4, q_1\rangle\}$	$\dfrac{s_4 \models \mu(q_1) \quad s_4 \to \{s_4\}}{s_4 \models \mu(q_0) = \Box(edb \to \Diamond(cert \vee fl))}$
Induction	$\mathcal{X}_5 = \{\langle s_1, q_0\rangle\}$, $Exit(\mathcal{X}_5) = \{\langle s_1, q_1\rangle\}$	$\dfrac{s_1 \models \mu(q_1) \quad s_1 \to \{s_1\}}{s_1 \models \mu(q_0) = \Box(edb \to \Diamond(cert \vee fl))}$
Successors	$\langle s_5, q_0\rangle$	$\dfrac{s_5 \to \{s_6, s_7\} \quad s_6 \models \mu(q_0) \wedge \mu(q_1) \quad s_7 \models \mu(q_0) \wedge \mu(q_1)}{s_5 \models \mu(q_0) = \Box(edb \to \Diamond(cert \vee fl))}$
Successors	$\langle s_2, q_0\rangle$	$\dfrac{s_2 \to \{s_3, s_4\} \quad s_3 \models \mu(q_0) \wedge \mu(q_1) \quad s_4 \models \mu(q_0) \wedge \mu(q_1)}{s_2 \models \mu(q_0) = \Box(edb \to \Diamond(cert \vee fl))}$
Successors	$\langle s_0, q_0\rangle$	$\dfrac{s_0 \to \{s_1, s_2\} \quad s_1 \models \mu(q_0) \wedge \mu(q_1) \quad s_2 \models \mu(q_0) \wedge \mu(q_1)}{s_0 \models \mu(q_0) = \Box(edb \to \Diamond(cert \vee fl))}$
Conclusion		$s_0 \models \mu(q_0) \Rightarrow s_0 \models \psi_3 \Rightarrow M \models \psi_3$

The proof can be started by showing first that the system trivially models the property in the states in which $\neg edb \vee \bigcirc\Diamond(cert \vee fl)$ holds (the fail axiom is applied). Starting from these states, where the model satisfies the property, by using first the induction rule then the successors rule, the proof traverses the automaton until it reaches the initial state. All the premises lead to conclude that it satisfies the $\mu(q_0)$ sub-formula. By construction of the proof [3], we can conclude that s_0 models the property, i.e., $M_{opt} \models \Box(edb \to \Diamond(cert \vee fl))$. According to the three-valued model checking algorithm, the result of the procedure is *maybe*, therefore the satisfaction of the property depends on the truth value that will be assigned to the proposition fl in the two uncertain states.

References

1. P. Arcaini, S. Bonfanti, A. Gargantini, A. Mashkoor, and E. Riccobene. Formal validation and verification of a medical software critical component. In *Formal Methods and Models for Codesign*, pages 80–89. IEEE, 2015.
2. A. Bernasconi, C. Menghi, P. Spoletini, L. D. Zuck, and C. Ghezzi. From model checking to a temporal proof for partial models. In *Software Engineering and Formal Methods*. Springer, 2017.

3. D. Peled and L. Zuck. From model checking to a temporal proof. In *International SPIN workshop on Model checking of software*, pages 1–14. Springer, 2001.