SMT-based Probabilistic Analysis of Timing Constraints in Cyber-Physical Systems

Li Huang* and Eun-Young Kang†
*School of Data & Computer Science, Sun Yat-Sen University, Guangzhou, China
huangl223@mail2.sysu.edu.cn
†PReCISE Research Centre, University of Namur, Belgium
eykang@fundp.ac.be

Abstract—Modeling and analysis of timing constraints is crucial in cyber-physical systems (CPS). EAST-ADL is an architectural language dedicated to safety-critical embedded system design. SIMULINK/STATEFLOW (S/S) is a widely used industrial tool for modeling and analysis of embedded systems. In most cases, a bounded number of violations of timing constraints in systems would not lead to system failures when the results of the violations are negligible, called Weakly-Hard (WH). We have previously defined a probabilistic extension of Clock Constraint Specification Language (CCSL), called PrCCSL, for formal specification of EAST-ADL timing constraints in the context of WH. In this paper, we propose an SMT-based approach for probabilistic analysis of EAST-ADL timing constraints in CPS modeled in S/S; an automatic transformation from S/S models to the input language of SMT solver is provided; timing constraints specified in PrCCSL are encoded into SMT formulas and the probabilistic analysis of timing constraints is reduced to the validity checking of the resulting SMT encodings. Our approach is demonstrated on a cooperative automotive system case study.

Index Terms—EAST-ADL, Timing Constraints, Probabilistic CCSL, SMT-based model checking, SIMULINK/STATEFLOW.

I. INTRODUCTION

Cyber-Physical Systems (CPS) are real-time embedded systems where the software controllers interact with physical environments. The continuous time behaviors of CPS often rely on complex dynamics as well as on stochastic behaviors. Modeling and analysis of timing constraints is essential to ensure the correctness of CPS. EAST-ADL is an architectural description language for safety-critical embedded systems design. The latest release of EAST-ADL has adopted the time model, which composes the basic timing constraints, i.e., repetition rates, end-to-end delays, and synchronization constraints. EAST-ADL relies on external tools, e.g., SIMULINK/STATEFLOW (S/S), for system behaviors description. SIMULINK (SL) is a block-diagram based formalism used to model continuous dynamics while STATEFLOW (SF) is used to specify control logic and state-based model behaviors of systems. Despite its strength in system modeling and simulation, S/S lacks of formal semantics to support rigorous verification of specifications. To tackle this shortcoming, efforts have been devoted into formal analysis of S/S models by using formal methods, e.g., model-checking, satisfiability modulo theory (SMT) solving. However, the conventional formal analysis of real-time systems addresses worst case designs, typically used for hard deadlines in safety-critical systems. The “Less-than-worst-case” models are far less investigated. In fact, in most cases, a bounded number of violations of timing constraints in systems would not lead to system failures when the results of the violations are negligible, called Weakly-Hard (WH) [1]. In this paper, we propose an SMT-based approach to support formal probabilistic analysis of EAST-ADL timing constraints in CPS modeled in S/S in the context of WH.

Clock Constraint Specification Language (CCSL) is a formal language for specification of both logical and dense timing constraints. We have previously defined a probabilistic extension of CCSL, called PrCCSL [2], which states that the relations (e.g., coincidence, causality and precedence) between events (e.g., input/output triggering, state changes) must hold with probability greater than or equal to a given probability threshold. Previous work is extended by including the supports of probabilistic analysis of timing constraints using SMT-based model checking: 1) S/S models, which describe the behaviors of systems, are transformed into the input language of SMT solver; 2) EAST-ADL timing constraints with stochastic properties are specified in PrCCSL and encoded into SMT formulas; 3) The probabilistic analysis of timing constraints is reduced into validity checking of the resulting SMT encodings. Our approach is demonstrated on a cooperative automotive system case study.

II. METHODOLOGY & EXPERIMENT

The overview of our approach is shown in Fig. 1. In our approach, S/S models are stored in ‘.mdl’ files, which contain textual descriptions of the compositions of the models. Z3 SMT solver [3] is employed as our verification engine. To trans-

Fig. 1. Overview of our approach

1EAST-ADL. https://www.maenad.eu/public/EAST-ADL-Specification_M2.1.9.1.pdf
2Simulink and Stateflow. https://www.mathworks.com/products.html
3Z3 SMT solver. https://github.com/Z3Prover/z3
late the stochastic functions (e.g., random number generation) in S/L, we adapt Z3PY (the Z3 API in Python) as encoding interface, in which the add-on modules for description of probability distributions can be leveraged. Given a system model in S/S and an EAST-ADL timing constraint $\phi$ (specified in PrCCL), the goal of our approach is to verify whether the probability of the constraint is greater than or equal to a probability threshold $p$, i.e., $Pr(\phi) \geq p$. To achieve this, we perform the following steps: 1) Extract necessary information (see Fig. 1) of S/S from .mdl file and translate S/S into Z3PY encodings based on the extracted information; 2) Encode PrCCL specifications of EAST-ADL timing constraints (ETC) in Z3PY and check the validity of the encodings using Z3.

**Translation of S/S into Z3PY:** Fig. 2 shows an excerpt of S/S model in .mdl file, in which each object (e.g., block, data or state) has a unique identifier named id. The data/variables in discrete-time S/S model are updated at sample time steps, which are translated into vectors (i.e., bounded lists) of appropriate sorts (e.g., integer, real and boolean). The index of the vectors represents the number of time steps that have proceeded during simulation. For instance, an integer signal $a$ is mapped to an integer list, with $a[i]$ ($i \in \mathbb{N}$) representing the value of signal $a$ at $i^{th}$ step during simulation. In S/L, lines are used for data transmission. During simulation, the data of ports connected by the same line are identical, which is interpreted as the equivalence of the data in Z3PY. The blocks of linear math/logic functions in S/L are mapped to the same arithmetic/logical operations in Z3PY straightforwardly.

![Fig. 2. S/S information in .mdl file](image)

*States* in S/F can be either active or inactive during simulation, declared as integer vectors whose elements are either 1 (active) or 0 (inactive) in Z3PY. The information of state in .mdl file can be divided into three classes (see Fig. 2): Hierarchy includes decomposition, history junction and the relation between superstates and their substates (indicated by treeNode). Transition represents the passage of the system from one state to another when the condition (i.e., a boolean expression) on the transition is true. State action refers to the operations (e.g., assignments) executed when the state is active, entered or exited. After the information of hierarchy, transitions and actions (HTA) is extracted from the .mdl file, the translation of S/F becomes the interpretation of HTA of each state in Z3PY, as presented in Algorithm 1.

**Algorithm 1: Translation of Stateflow into Z3PY**

```
Input: Simulation bound N, states $s_1, \ldots, s_j, \ldots, s_n$ ($s_j$ is the state with id $j$), information of hierarchy, actions and transitions of each state;
Output: $E$: the Z3PY encodings of Stateflow chart;
1 $E \leftarrow \emptyset$
2 for $j = 0; j < n; j + +$ do
3     $s_j \leftarrow$ IntegerVector($N$)
4     if $s_j$.substate $\neq \emptyset$ then
5         $E \leftarrow E \cup$ Encode($s_j$.hierarchy)
6     if $s_j$.action $\neq \emptyset$ then
7         $E \leftarrow E \cup$ Encode($s_j$.action)
8     if $s_j$.transition $\neq \emptyset$ then
9         $E \leftarrow E \cup$ Encode($s_j$.transition)
10 return $E$
```

**III. CONCLUSION AND FUTURE WORK**

We present an SMT-based approach to perform probabilistic analysis of EAST-ADL timing constraints in CPS described in Simulink/Stateflow. The practicality of our approach is demonstrated on a CAS case study. As ongoing work, the application of our approach in larger-scale case studies will be investigated and an automatic translator from Simulink/Stateflow (.mdl file) to Z3PY will be developed.

**REFERENCES**

[1] G. Bernat, A. Burns, and A. Llamosi, “Weakly hard real-time systems,” Transactions on Computers, vol. 50, no. 4, pp. 308 – 321, 2001.

[2] E. Kang and L. Huang, “Probabilistic analysis of timing constraints in autonomous automotive systems using Simulink Design Verifier,” in SETTA, 2018.

[3] E. Kang, L. Huang, and D. Mu, “Formal verification of energy and timed requirements for a cooperative automotive system,” in SAC. ACM, 2018, pp. 1492 – 1499.