Using Machine Learning to Minimize User Intervention in Theorem Proving based Dynamic Fault Tree Analysis

Yassmeen Elderhalli, Osman Hasan, and Sofiène Tahar

Concordia University, Montreal, Quebec, Canada
{y_elderh,o_hasan,tahar@ece.concordia.ca}

Dynamic fault trees (DFTs) have become one of the commonly used modeling techniques that capture the dynamic failure behavior of systems. Recently, DFTs have been formalized in higher-order logic (HOL), which allows performing DFT analysis within the sound core of a HOL theorem prover. However, due to the interactive nature of HOL theorem proving, the proof process involves significant user guidance. In this paper, we propose to use machine learning techniques to facilitate automating the proof of the subgoals. The machine learning can use the existing proofs of these goals as well as the verification steps being performed at runtime to come up with reasoning to verify the remaining subgoals. This kind of support from machine learning can lead to the creation of a tool for DFT analysis that requires minimum user intervention in the formal DFT analysis and thus can facilitate the industry to benefit from a sound DFT analysis approach.

Dynamic Fault Trees A dynamic fault tree (DFT) is a graphical representation of the sources of system failure in a tree format [12]. The system modeling starts with a top event that represents an undesired failure event, then the sources that lead to the occurrence of this top event are modeled using fault tree gates. The importance of DFTs lies in the fact that they can capture the failure dependencies among system components using DFT gates, which is suitable for analyzing real-world systems specially the safety-critical ones.

Formal DFT Analysis Giving the safety-critical nature of the applications of DFT analysis, we have recently provided the formalization of DFT in HOL4 [7], which allows conducting the DFT analysis in a sound theorem prover (TP) [4, 3] based on the algebraic approach [9]. However, due to the interactive nature of HOL TP, using our formalization for DFT analysis is limited to users with a considerable experience in HOL TP. Our formalization is based mainly on verifying the probabilistic behavior of DFT gates and utilizing the probabilistic principle of inclusion and exclusion (PIE) to express the probability of the top event of a given DFT. The number of subgoals to verify a given DFT depends on the number of subevents to be included in the PIE. For example, if the PIE is used with 7 subevents, it is required to verify 127 different subgoals. We have verified several intermediate lemmas that facilitate the proof process of DFT case studies. Moreover, we have identified certain patterns in the proof process of these lemmas that enable extending them to cover larger case studies. However, so far there is no automation involved in this process, as the generated subgoals are different since they represent the different combinations of intersection of the subevents of the PIE.

Using Machine Learning in Formal DFT Analysis In this project, we propose to use machine learning to facilitate automating the proofs of DFT analysis. These proofs deal mainly with iterated Lebesgue integrals and their measurability. Although these proofs are complex, there are some common patterns that can be utilized in the automation process. This automation enables using this formalization by other users that are not experts in TP. Ultimately, we plan to develop a tool that would provides the user with an interface for conducting the analysis without getting involved into the details of the proofs.

Related Work Machine learning (ML) is used with automated TP to select the heuristic for proof search based on features of the conjecture to be proved [2]. Recently, ML has been proposed to be used with HOL TP. For example, in [8], a dataset is created for the proof steps based on the multivariate theory and the proof of the Kepler conjecture [6] in HOL Light TP [1]. This dataset and similar ones can be utilized in classifying useful steps in proving a certain conjecture.

Proposed Methodology We propose to create a similar dataset for the proof steps of the measure, Lebesgue integral and probability theories [10, 11] of HOL4 TP, since these theories form the basis of our proofs for DFTs, as depicted in Figure 1. In this dataset, we will also include our formalization for DFT, particularly the intermediate lemmas that have certain patterns in their proof steps. This dataset will be divided into two subsets; a training and a testing set. Based on the created set, we plan to use ML in the premise selection of the proper verified theorems that can be helpful in verifying a given conjecture. This is basically a classification problem of whether a certain theorem is helpful or not for proving the current conjecture. Therefore, including

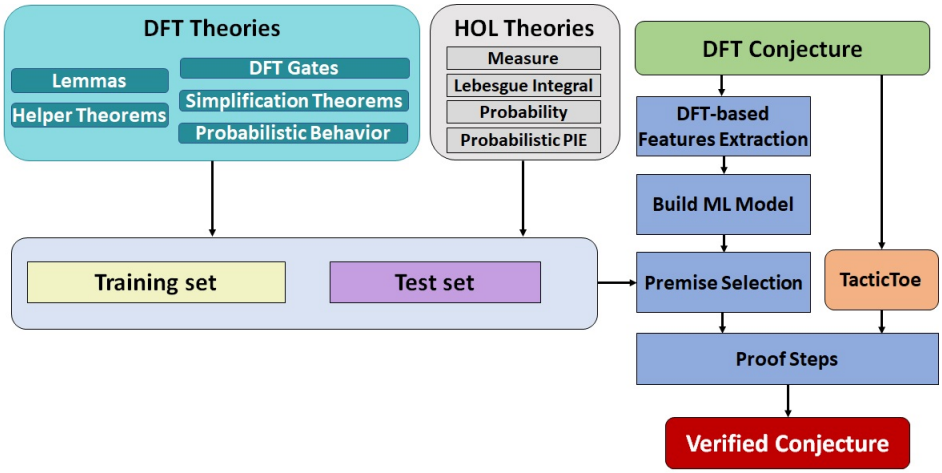

Figure 1: Proposed Methodology

our intermediate lemmas of DFT analysis in the training set is of a great importance, as the patterns that we identified can facilitate determining the right steps and theorems to verify similar conjectures. This task requires creating an ML model, such as convolutional neural networks [13], in order to classify the theorems. We plan to consider features related to DFT gates that can further help in the classification process, as the proofs that we developed differ depending on the DFT gate. In addition, we will make use of the TacticToe approach implemented in [5] that automates the selection of the proper tactics to prove a conjecture in HOL4.

After successfully implementing the above-mentioned steps, we plan to enhance the learning process by enabling learning on the fly, i.e., continuously enabling learning while conducting new proofs. This step will positively impact the classification results as the training set will be continuously improved with each new proven conjecture. We believe that implementing this automation for formal DFT analysis will allow end users that are unfamiliar with TP to benefit from our DFT formalization to provide sound analysis.

References

[1] HOL-light Theorem Prover, http://www.cl.cam.ac.uk/ jrh13/hol-light//, 2018.

[2] J. Bridge. *Machine Learning and Automated Theorem Proving*. PhD thesis, University of Cambridge, UK, 2010.

[3] Y. Elderhalli, W. Ahmad, O. Hasan, and S. Tahar. Formal Probabilistic Analysis of Dynamic Fault Trees in HOL4. Tech. rep., Concordia University, Canada. https://arxiv.org/abs/1807.11576, 2018.

[4] Y. Elderhalli, O. Hasan, W. Ahmad, and S. Tahar. Formal Dynamic Fault Trees Analysis Using an Integration of Theorem Proving and Model Checking. In *NASA Formal Methods*, LNCS 10811, pages 139–156. Springer, 2018.

[5] T. Gauthier, C. Kaliszyk, and J Urban. TacticToe: Learning to reason with HOL4 Tactics. In *Logic for Programming, Artificial Intelligence and Reasoning*, volume 46, pages 125–143, 2017.

[6] T. C. Hales, J. Harrison, S. McLaughlin, T. Nipkow, S. Obua, and R. Zumkeller. A Revision of the Proof of the Kepler Conjecture. *Discrete & Computational Geometry*, 44(1):1–34, 2010.

[7] HOL4. https://hol-theorem-prover.org/, 2018.

[8] C. Kaliszyk, F. Chollet, and C. Szegedy. Holstep: A machine learning dataset for higher-order logic theorem proving. *arXiv preprint arXiv:1703.00426*, 2017.

[9] G. Merle. *Algebraic Modelling of Dynamic Fault Trees, Contribution to Qualitative and Quantitative Analysis*. PhD thesis, ENS, France, 2010.

[10] T. Mhamdi, O. Hasan, and S. Tahar. On the Formalization of the Lebesgue Integration Theory in HOL. In *Interactive Theorem Proving*, LNCS 6172, pages 387–402. Springer, 2010.

[11] T. Mhamdi, O. Hasan, and S. Tahar. Formalization of Entropy Measures in HOL. In *Interactive Theorem Proving*, LNCS 6898, pages 233–248. Springer, 2011.

[12] E. Ruijters and M. Stoelinga. Fault Tree Analysis: A Survey of the State-of-the-art in Modeling, Analysis and Tools. *Computer Science Review*, 15-16:29 – 62, 2015.

[13] J. Schmidhuber. Deep learning in Neural Networks: An Overview. *Neural networks*, 61:85–117, 2015.