Reimplementing and Evaluating Static Fault Tree Analysis using BDDs

Daniel Basgöze

Software Modeling and Verification, RWTH Aachen University, Aachen, Germany

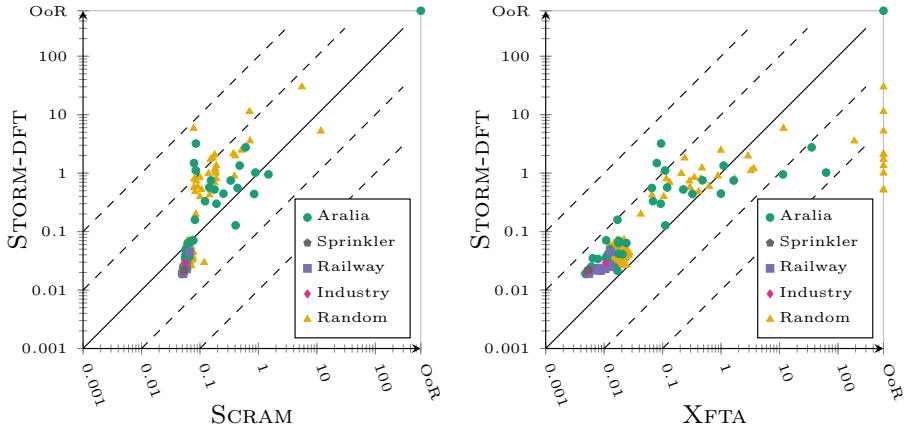
Fault Tree analysis (FTA) is a popular and established method to model and assess risk in complex systems such as nuclear power plants or airplanes [10,11]. Traditional *Static Fault Trees* (SFT) represent a monotonic Boolean function that models the overall failure state of a system depending on the failure states of its components. The most common approach of analyzing them is to first translate the fault tree into a *binary decision diagram* (BDD) and analyzing the BDD instead [8,11]. We present a comparison of our implementation of BDD-based FTA in the probabilistic model checker Storm [1,2,6] with other state-of-the-art academic FTA tools Scram [7] and Xfta [9]. Furthermore, we provide an artifact of the experimental evaluation of all three tools that includes patches, installation and evaluation scripts, tool configurations, and fault tree models[1].

The static fault tree analysis implementation in Storm is based on the multi-core BDD library Sylvan [3] and was a result of our efforts to speed up the analysis of *dynamic fault trees* (DFT) [4] in Storm-dft [13]. This was done by reimplementing a modular analysis approach that can analyze some of the static parts of a DFT with traditional BDD-based methods [5]. The resulting SFT analysis implementation is fully fledged and supports calculating *minimal cut sets, unreliability,* and *importance measures.*

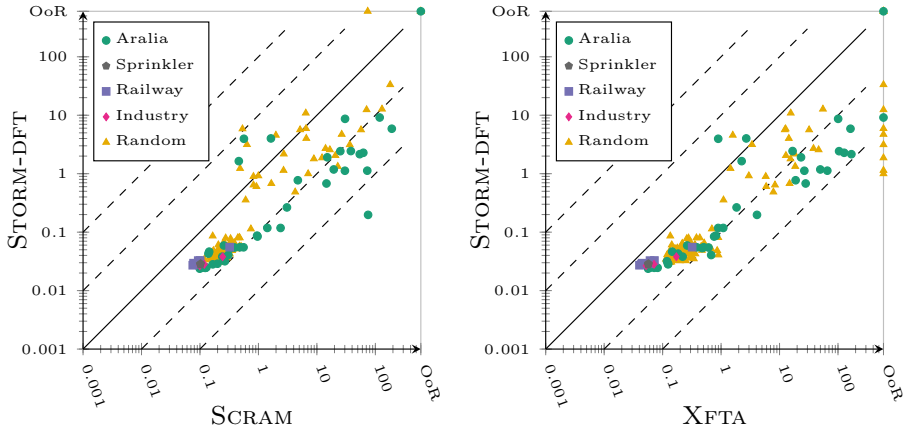
We evaluated the performance of Storm-dft, Scram, and Xfta on 215 examples from 5 benchmark sets. Fig. 1(a) compares the runtimes of Storm-dft against Scram and Xfta when computing the failure probability for a single time point. In comparison Fig. 1(b) compares the runtimes when computing the failure probabilities for 10 000 time points. Our evaluation showed that Storm-dft is competitive with the other SFT analysis tools. Further our implementation is significantly faster when calculating the unreliability for multiple time points as it exploits vectorization.

One problem we encountered during our evaluation is that the current version of Scram found in its official Github repository is not maintained anymore and does not build without some changes to its source code. We therefore provide the necessary patch in our artifact. While Scram and Xfta both state support for the XML-based *Open-PSA Model Exchange format* [12] some differences remain. For example Scram uses the keyword 'system-mission-time' where Xfta uses 'mission-time'. Storm-dft meanwhile uses the Galileo file format[2]. To deal with this we developed and published a translation script in our artifact that can convert between these different formats.

[1] https://doi.org/10.5281/zenodo.5834213
[2] https://dftbenchmarks.utwente.nl/galileo.html



(a) Calculation for one time point



(b) Calculating for 10 000 time points

Fig. 1. Runtime comparisons for computation of unreliability. *OoR* indicates *out of resources* and represents either a timeout (5 min) or memory out (30 GB).

References

1. Basgöze, D.: Dynamic fault tree analysis using binary decision diagrams (2020), Bachelor thesis, RWTH Aachen University
2. Basgöze, D., Volk, M., Katoen, J.P., Khan, S., Stoelinga, M.: Bdds strike back: Efficient analysis of static and dynamic fault trees (2022)
3. van Dijk, T.: Sylvan: multi-core decision diagrams. Ph.D. thesis, University of Twente, The Netherlands (2016)
4. Dugan, J.B., Bavuso, S.J., Boyd, M.A.: Fault trees and sequence dependencies. In: RAMS. pp. 286–293 (1990)
5. Gulati, R., Dugan, J.B.: A modular approach for analyzing static and dynamic fault trees. In: RAMS. pp. 57–63. IEEE (1997)

6. Hensel, C., Junges, S., Katoen, J.P., Quatmann, T., Volk, M.: The probabilistic model checker storm. Int. J. Softw. Tools Technol. Transf. pp. 1–22 (2021)
7. Rakhimov, O.: Scram probabilistic risk analysis tool (2018). https://doi.org/10.5281/zenodo.1146337
8. Rauzy, A.: New algorithms for fault trees analysis. Reliab. Eng. Syst. Saf. **40**(3), 203–211 (1993)
9. Rauzy, A.: Probabilistic safety analysis with XFTA. AltaRica Association (2020)
10. Ruijters, E., Stoelinga, M.: Fault tree analysis: A survey of the state-of-the-art in modeling, analysis and tools. Comput. Sci. Rev. **15**, 29–62 (2015)
11. Stamatelatos, M., Vesely, W., Dugan, J., Fragola, J., Minarick, J., Railsback, J.: Fault Tree Handbook with Aerospace Applications. NASA Washington, DC (2002)
12. Steven, E., Antoine, R.: Open-PSA Model Exchange Format. The Open-PSA Initiative (2007)
13. Volk, M., Junges, S., Katoen, J.: Fast dynamic fault tree analysis by model checking techniques. IEEE Trans. Ind. Informatics **14**(1), 370–379 (2018)