

# FOSSIL: A Software Tool for the Formal Synthesis of Lyapunov Functions and Barrier Certificates using Neural Networks

Alessandro Abate<sup>1</sup>, Daniele Ahmed<sup>2</sup>, Alec Edwards<sup>3</sup>, Mirco Giacobbe<sup>4</sup> and Andrea Peruffo<sup>5</sup>

## Keywords

Lyapunov functions, Barrier Certificates, Neural Networks, SAT Modulo Theory

We propose an automatic and formally sound method for synthesising Lyapunov functions for the asymptotic stability, as well as Barrier certificates (or functions) for the safety analysis, of autonomous non-linear dynamics represented as systems of ordinary differential equations.

Typical methods are either analytical and require manual effort or are numerical but lack of formal soundness. Symbolic computational methods instead, give formal guarantees but are typically semi-automatic because they rely on the user to provide appropriate function templates.

We propose a method that finds Lyapunov functions or Barrier certificates automatically—using machine learning—while also providing formal guarantees—using satisfiability modulo theories (SMT). We employ a counterexample-guided inductive synthesis (CEGIS) approach where a numerical learner and a symbolic verifier interact to construct provably correct Lyapunov neural networks (LNNs). The learner trains a neural network that seeks to satisfy given sufficient criteria for asymptotic stability or safety over a set data points sampled from the state space; the verifier proves via SMT-solving that the criteria are satisfied over the specified domain or augments the data set with counterexamples.

This work is bolstered by a software tool, FOSSIL [1]. The tool has two core contributions on synthesis—automation and soundness—both of which are attained by means of the inductive, counter-example-based method. This method exploits the flexibility of candidate functions generated by training neural network templates, the formal assertions provided by the verifier, and finally new procedures to ease the exchange of information between the two mentioned components.

In particular, FOSSIL combines the neural network templates with an enhanced CEGIS loop, depicted in Figure 1. As input, the procedure requires both the dynamical system in question and the desired specification to certify: either stability—leading to Lyapunov function synthesis—or safety, requiring synthesis of a barrier certificate. The loop contains the fundamental components within CEGIS: namely a learner, previously specified as a neural network, while an SMT-solver


---

OVERLAY 2021: 3rd Workshop on Artificial Intelligence and Formal Verification, Logic, Automata, and Synthesis, September 22, 2021, Padova, Italy

✉ alessandro.abate@cs.ox.ac.uk (A. Abate); alec.edwards@cs.ox.ac.uk (A. Edwards); mirco.giacobbe@cs.ox.ac.uk (M. Giacobbe); andrea.peruffo@cs.ox.ac.uk (A. Peruffo)



© 2021 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

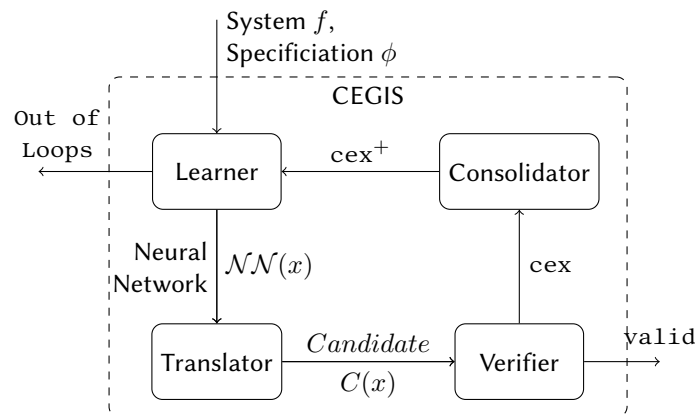
takes on the role of the verifier. In addition to these two base components, FOSSIL augments the CEGIS architecture with two additional components, known as the translator and the consolidator. The task of these intermediary components is described as follows.

Firstly, the translator acts as an intermediary from the learner to the verifier by seeking to convert the numerical neural network to a symbolic candidate solution. This process involves a rounding operation on the corresponding weights matrices and biases of the neural network in order to simplify the expressions passed to the verifier and aid the consequent verification step. In contrast, the consolidator sits within the communication channel from verifier to learner and seeks to augment any counterexample obtained by the verifier, so as the ‘consolidate’ the subsequent learning achieved by the learner. This is achieved with two key observations: further counter-examples likely exist near that given by the SMT-solver, which can be found by sampling from the state-space around this original point. Secondly, the functions provided to the verifier as candidate solutions are almost-everywhere differentiable, allowing for the traversing of these functions using gradient ascent (or descent). This in turn obtains further states along the candidate solution which invalidate the required conditions, and can be provided to the learner for further training. The improved communication achieved using the enhanced CEGIS loop aids both the robustness and speed of synthesis.

The work presented builds on previous works involving the use of a counter-example guided approach to synthesis of Lyapunov functions [2]; [3] then incorporates the use of neural network templates for these functions before being extended to the synthesis of barrier certificates in [4].

## References

- [1] A. Abate, D. Ahmed, A. Edwards, M. Giacobbe, A. Peruffo, FOSSIL: A Software Tool for the Formal Synthesis of Lyapunov Functions and Barrier Certificates Using Neural Networks, Association for Computing Machinery, New York, NY, USA, 2021. URL: <https://doi.org/10.1145/3447928.3456646>.



**Figure 1:** Enhanced CEGIS loop as part of the modular architecture within FOSSIL.

- [2] D. Ahmed, A. Peruffo, A. Abate, Automated and Sound Synthesis of Lyapunov Functions with SMT Solvers, in: TACAS (1), volume 12078 of *LNCS*, Springer, 2020, pp. 97–114.
- [3] A. Abate, D. Ahmed, M. Giacobbe, A. Peruffo, Formal Synthesis of Lyapunov Neural Networks, *IEEE Control Systems Letters* 5 (2021) 773–778.
- [4] D. Ahmed, A. Peruffo, A. Abate, Automated formal synthesis of neural barrier certificates for dynamical models, in: TACAS, volume 12651 of *LNCS*, Springer, 2021, pp. 370–388.