Neuro-Symbolic Lemma Conjecturing

Sólrún Halla Einarsdóttir, Chalmers University of Technology

CHALMERS UNIVERSITY OF TECHNOLOGY

Dept of Computer Science and Engineering

email: slrn@chalmers.se

Can AI discover interesting mathematical lemmas?

Theory exploration is the automatic discovery of interesting conjectures and lemmas about mathematical objects.
For example, given definitions of addition (+) and multiplication (*), a theory exploration tool might conjecture properties such as **x + y = y + x** and **a * (b + c) = a * b + a * c**.
If the tool is equipped with automated proof methods it can also find proofs for these conjectures so that they can be used as valid lemmas. Such lemmas can be used to strengthen automated theorem provers, or in an interactive setting to aid a human user in their mathematical formalization. Previously, we have developed symbolic tools for theory exploration which have been used to successfully discover, for example, lemmas needed in automated (co)-inductive provers **[2,5]**.
In light of the recent impressive results achieved by Large Language Models (LLMs) in various text-generation tasks, we want to examine how LLMs can be used for lemma generation in a theory exploration setting, and how they can be combined with symbolic tools for optimal results.

A neuro-symbolic approach

LLMs are remarkably good at learning patterns from their training data and generating output that fits a similar pattern for a given query context. Therefore, they can potentially be trained to generate lemmas similar to those previously seen for mathematical definitions analogous to those given, if exposed to the right kind of training data. A weakness of neural models such as LLMs is that they may be prone to generating repetitive or redundant lemmas and fail to discover more novel and useful lemmas.
Another flaw that must be addressed when using LLMs in this context is the fact that there are no correctness guarantees on the LLM's output, so the generated lemmas may simply be false.
Unlike neural methods, symbolic methods can be designed and programmed to generate only true statements and avoid repetition and redundancy. However, symbolic methods will only generate lemmas that fit a predefined specification from within a specified search space, and tend to scale poorly to a larger search space.
To address these shortcomings, we propose a neuro-symbolic lemma conjecturing tool with the following implementation: An LLM is trained to generate lemma templates that describe the shape of a lemma rather than generating complete lemmas. Then symbolic methods are used to fill in the details. In this way, we leverage the best of both neural and symbolic methods, using the LLM to capture the intuition and suggest appropriate patterns and symbolic methods to ensure correctness and novelty. As far as we are aware, this is the first work focusing on neuro-symbolic conjecturing of novel lemmas. With the great success LLMs have displayed in generative tasks, it is crucial to examine the potential ways to use them in combination with reliable symbolic methods for optimal results and efficiency.

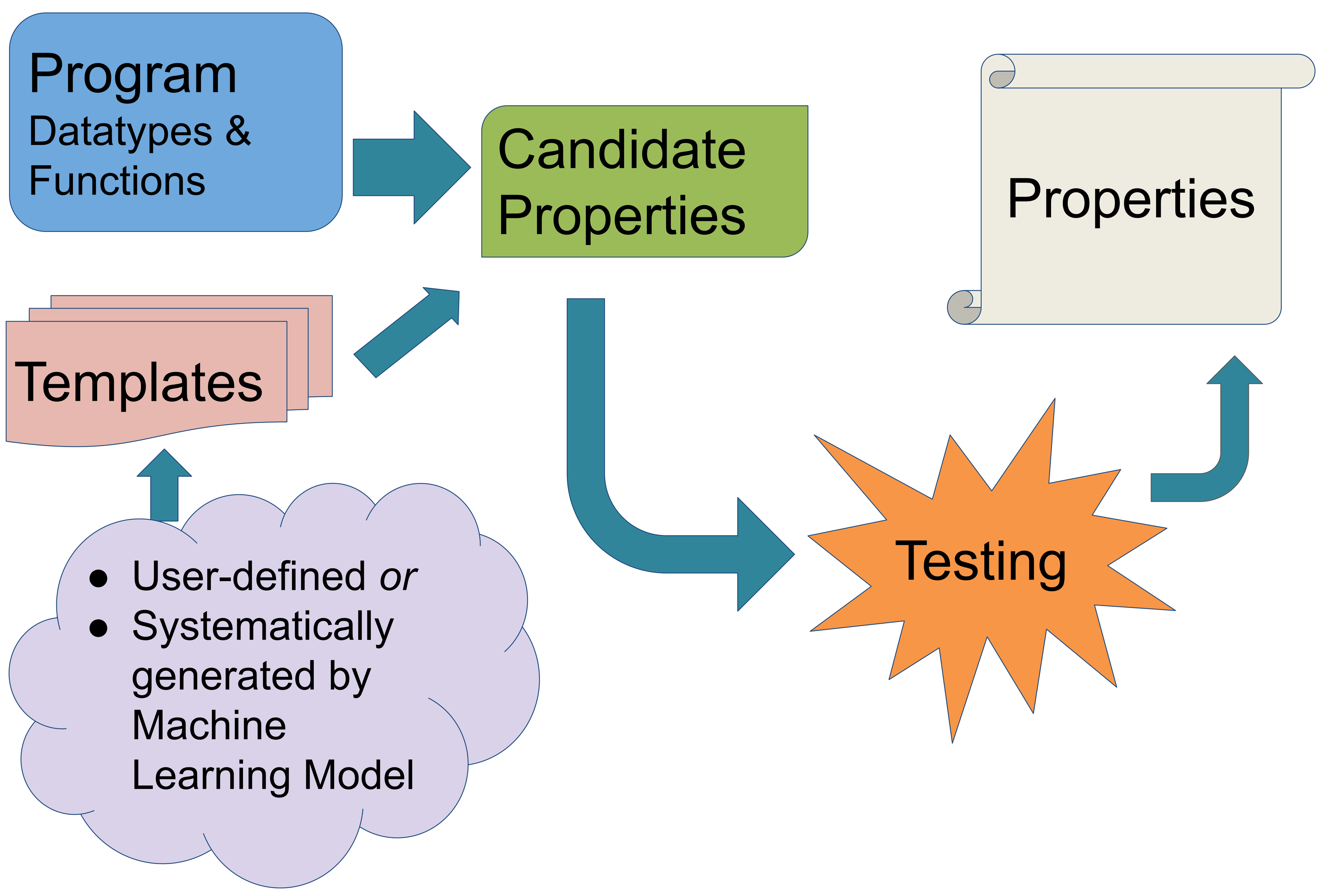
References

1. SH Einarsdóttir et al, *On Lemma Conjecturing using Neural, Symbolic and Neuro-symbolic approaches,* AITP 2024.
2. SH Einarsdóttir et al, *Lemma Discovery and Strategies for Automated Induction,* IJCAR 2024.
3. SH Einarsdóttir, *Theory Exploration for Programs and Proofs,* Licentiate Thesis, Chalmers University of Technology, 2022.
4. SH Einarsdóttir et al, *Template-Based Theory Exploration: Discovering Properties of Functional Programs by Testing,* IFL 2020.
5. SH Einarsdóttir et al, *Into the Infinite-Theory Exploration for Coinduction,* AISC 2018.

Research Questions addressed in ongoing work

1. Can an LLM be trained to generate useful **lemmas** for a given set of function definitions? How does this approach compare to existing symbolic tools?
2. Can an LLM be trained to generate useful **lemma templates** to be filled in symbolically using a tool like RoughSpec? How does this approach compare to the one above? How does it compare to using RoughSpec with a set of standard templates?
3. What level of contextual information is useful for an LLM to generate lemmas and lemma templates?

RoughSpec: a template-based theory exploration tool [3]



WASP WALLENBERG AI, AUTONOMOUS SYSTEMS AND SOFTWARE PROGRAM