

Open Problems in Formal Artificial Intelligence

Dustin Juliano

OPLSS 2025

2025-06-24

Overview

- Opaque AI models can be thought of as obfuscated computer programs and data in one or more (possibly nested) domain-specific languages.
- The journey to Formal AI begins by making this observation an explicit construction with additional constraints.

Overview

- Formal AI *constructs* (compare with "trains") AI *solutions* that are, ideally, correct-by-construction.
- A solution is a collection of one or more well-formed, *human-readable* computer programs + data.
- The data portion of the solution must not be used to avoid the human-readability constraint.
- In Formal AI there is no distinction between solutions (which include models) and their descriptions; they are the same object.

Motivation

- Control

- It is difficult to control AI we do not fully understand.
- Formal AI builds a bridge to existing work in PL/TT.
- We gain the ability to *prove* that an AI does exactly as we intend.

- Modularity

- We will be able to pre-construct (pre-train), compose, and reuse AI features in the form of libraries.
- Imagine a global library of reusable AI capabilities that can simply be "imported" into a new AI solution.

- Efficiency

- Not restricted to a single search/optimization algorithm.
- Can target custom programming languages and hardware.
- Can use optimal algorithms and data structures for sub-solutions.

Challenges

- Search Space Reduction
 - Target languages
 - Programming Languages
 - Data Languages
 - Intermediate Representations
- (Meta-)heuristics & Search
 - Discrete optimization, combinatorial optimization, evolutionary algorithms, and so on.
 - Finding a replacement for SGD and its variants.
 - Important: Not restricted to single approach/algorithm!
- Highly Optimized Implementations
 - It *has* to be fast because a large number of permutations/iterations will be required.

Challenges

- Labeling Problem
 - What patterns can and should be used?
 - How to define those patterns?
 - Extending to comments (possibly requiring advanced AI).
- Legibility Requirement
 - Two-sided:
 - How to find more solutions that would be ruled out by legibility.
 - How to meet and extend legibility.
- Scaling Comprehensibility
 - How to keep up with continuously improving Formal AI solutions that require larger and larger groups of people to understand.

First Steps

- Systems level language with dependent types
 - A first target language for expressing Formal AI solutions.
 - Is a useful parallel goal that will be applicable outside Formal AI.
- Data description languages with rich specifications/types
 - Designed to work with the target programming language(s).
 - Must be highly efficient, as it must also be incrementally modified by the Formal AI solver.
- Generative Programming + Program Synthesis

Thank You

`djuliano@iu.edu`

`juliano.author@gmail.com`

`https://aisecurity.org`

`https://formalai.org`

`https://formalai.dev`