EDGE CASE

Embracing Change: LLM Use in Safety Engineering

Michael Wagner, Chief Safety Officer, Edge Case
mwagner@ecr.ai

The ideas in this presentation are drawn from numerous conversations, including those with Anoop Balakrishnan, Ben Hocking, Jonathan Rowanhill, Aaron Kane, Justin Ray, Jacob Nelson, and many others. I'm grateful for their thoughts and their time.

Claim: Artificial intelligence, especially LLMs, shows great promise for building safer, more dependable systems.

What about hallucinations?

What about a lack of explainability?

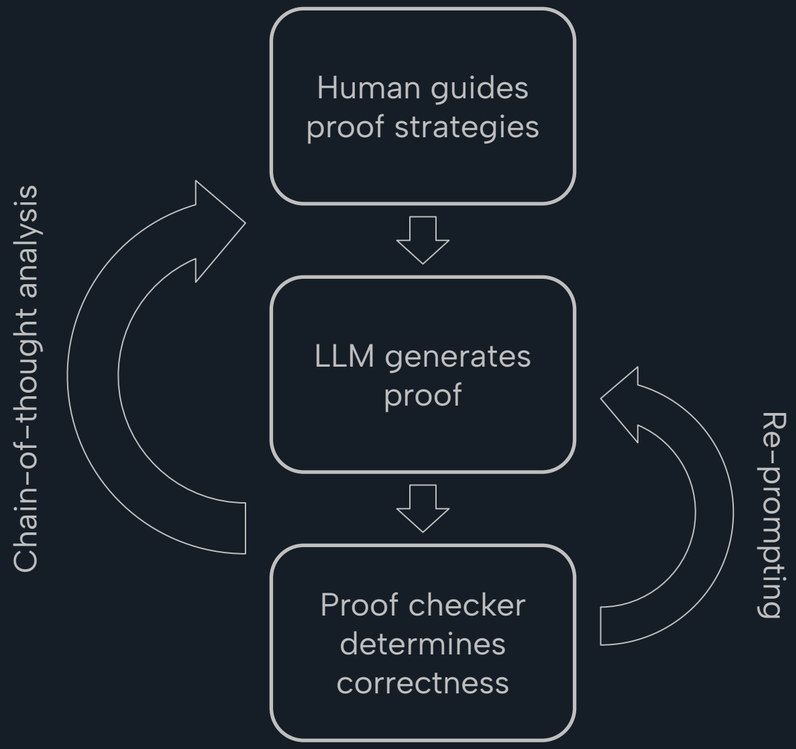
What about cultural challenges?

Pitfalls of Putting a Human in the Loop

- What can an LLM produce?
- Everything! Specifications, software implementations, hazard analyses, official reports, …
- LLM output is voluminous are efficiency gains lost if humans review?
- LLM output is convincing humans will be fooled
- Irony of automation reduction of human attention & expertise
- Problems that need an ML solution are too complex for human-checked “white box” analysis
- Chain-of-thought is no more trustworthy than generated content!

LLMs, Formal Methods, and Autoformalization

- Proof construction is difficult, even for experts…
  - …but proof **checking** is easy for theorem checkers.
- Trustworthy tools can check proofs proposed by untrustworthy (but efficient) LLMs
- In his work on “autoformalization”, Szegedy expands this simple idea into a strategy for AI that produces verifiable and explainable outputs



A. Kozyrev, G. Solovey, N. Khramov, A. Podkopaev, "CoqPilot, a plugin for LLM-based generation of proofs" (2024)
K. Yang et al., "LeanDojo: Theorem Proving with Retrieval-Augmented Language Models" (2024)
Y. Wu et al., “Autoformalization with Large Language Models” (2022)

Can LLMs Make Formal Methods Easier to Use?

- LLMs are good at converting between natural & formal languages
- Allows formal methods practitioners to use their language of choice
- Allows practitioners to more easily learn additional formal languages
- Could apply multiple languages to the same project
  - TLA+ for temporal aspects, PVS for safety invariants, etc.
- Flaws in the specification that are difficult to identify in one language might become more obvious in another
- Overreliance on tools can undermine critical thinking by humans

Credit: Ben Hocking

Case Study 1 Question:
"Prove that there is no function f from the set of non-negative integers into itself such that $f(f(n)) = n + 1987$ for every n."

Codex Output:

```
theorem
  fixes f :: "nat \<Rightarrow> nat"
  assumes "\<forall> n. f (f n) = n + 1987"
  shows False
```

(Yu et al., 2022)

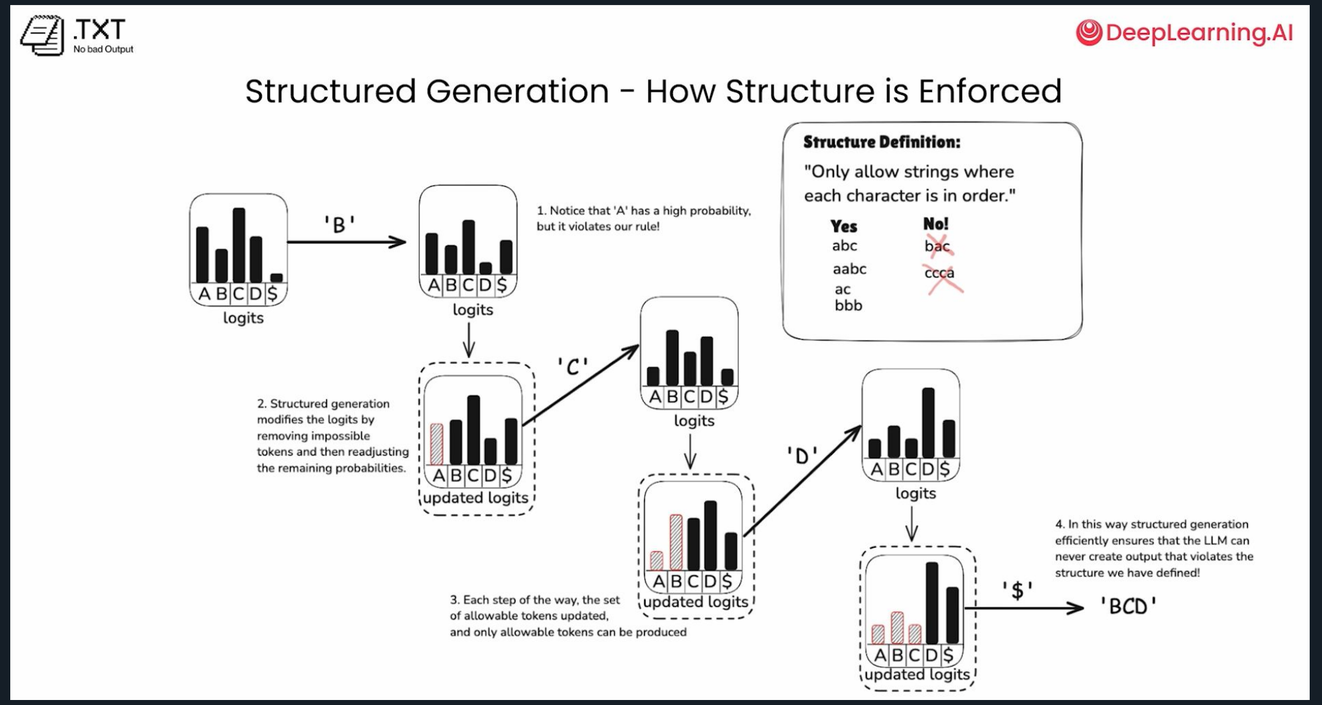
Extending the Strategy

- Specification mining from source code & dynamic analysis is very common
- Does not require LLMs

- However, there seems to be a dearth of research into autoformalization from SysML models
- Should be more efficient & effective than starting with raw natural language as long as MBSE process is disciplined (avoid free text annotations!)

Structured Generation

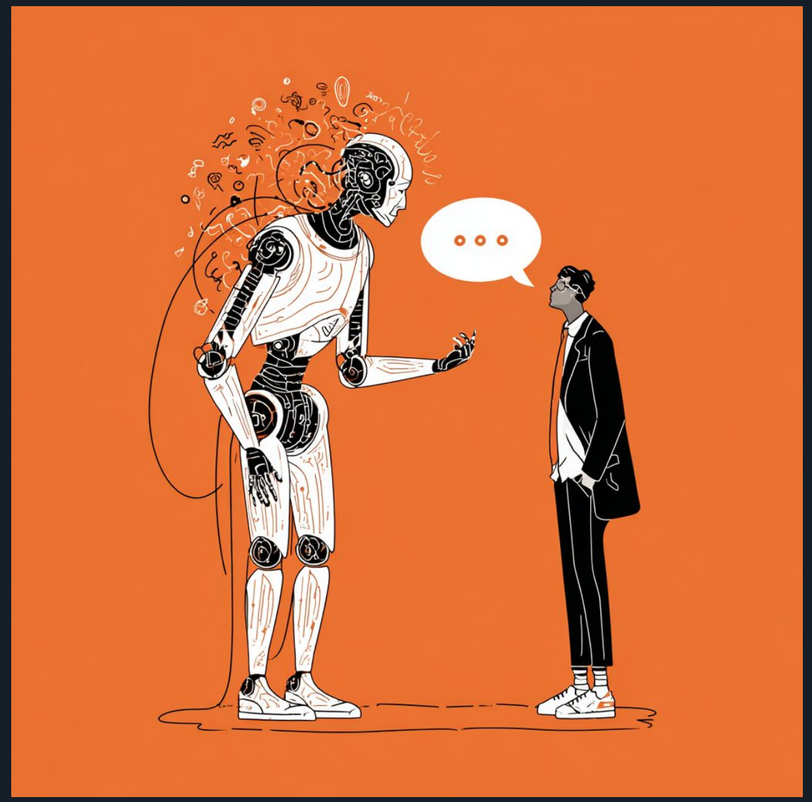
- LLM inference generates outputs within defined “spaces” like English grammar
- Model architecture can be adjusted to enforce constraints within this space, such as regex called **structured generation**
- Can reduce retries required to generate valid proofs but can the approach be extended with formal syntax & invariants?



Credit: DeepLearning.AI

Returning to the Human in the Loop

- Where trustworthy* formal verification is possible, allow LLM to lead artifact generation
- In all other cases, think of the LLM like a very flexible **linter**
- An LLM can understand data across disconnected tools (see “Safety Factories”!)
- Scalable review of human-generated artifacts improve quality and reduce iterations of reviews
- Better verification plans



** Caution: AI may optimize around bugs in proof checkers!*

Challenge: Validating Driving Models

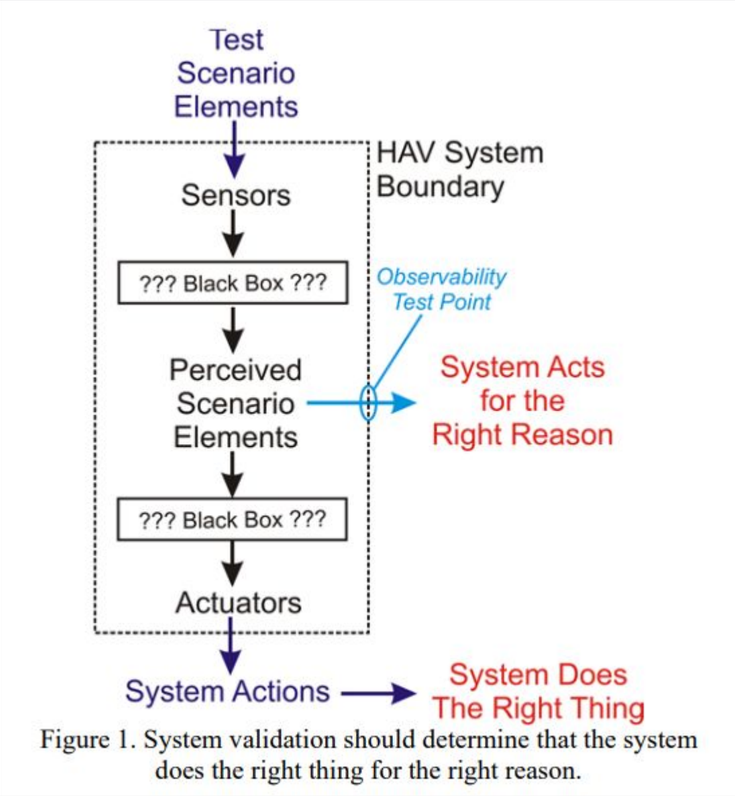


Figure 1. System validation should determine that the system does the right thing for the right reason.

(Koopman & Wagner, 2018)

Credit: Wayve & NVIDIA

✓ Open loop!

✗ Closed loop!

Safety gates & fail-safes

Safety performance indicators

Statistical behavioral validation

EDGE CASE

Thank You!

Michael Wagner, Chief Safety Officer, Edge Case
mwagner@ecr.ai