The Opportunity for AI and Formal Verification

Atlas Computing is showing that a new architecture could allow and require AI to provide verifiable output. This document describes how this architecture could secure cyber systems, as part of our overall strategy.

Executive Summary

Today, critical infrastructure is vulnerable to both malicious attacks and unintended failures, and these risks are expected to grow in the foreseeable future. Deploying formal verification (FV) across critical cyber physical systems would dramatically improve safety and security, but has historically been too costly to use outside the simplest or most critical subsystems. AI could allow widespread use of FV in years not decades, shifting cyber risks strongly in favor of defense. To achieve this, we need to advance AI-assisted FV capabilities, grow the FV talent funnel, and integrate FV into strategies for enhancing national security and human rights in the face of AI advancements. Atlas Computing is helping to advance FV by supporting research, tech transfer, and startup formation. This creates opportunities for researchers, policymakers, tech companies, infrastructure operators, entrepreneurs, and investors to collaboratively define better AI tools for the future.

Strengthening Cyber Physical Infrastructure Is Critical

The software running the modern world is vulnerable. The struggle to protect American infrastructure sees frequent escalations[1], with FBI Director Christopher Wray recently testifying that "China's hackers are positioning on American infrastructure in preparation to wreak havoc"[2]. This risk ranges from power plants and cars to pacemakers and routers.

AI architectures, optimized for tasks like text completion, now answer programming challenges nearly as well as median software engineers,[3] which enables products like GitHub's Copilot. One experiment showed Copilot reduced the time needed to complete a software project by 55%[4], with one user quoted saying that, with Copilot, "I have to think less, and when I have to think it's the fun stuff." Ensuring security is rarely "the fun stuff." Furthermore, a 2022 study found that participants with access to an AI assistant wrote "significantly less secure code than those without access"[5], and it's not guaranteed this trend will reverse as assistants improve.

As research advances, we may see AI code generators suddenly able to patch vulnerabilities and generate exploits simultaneously, leading to an arms race that favors attackers, since patches take time to deploy. Even before that point, AI code generators will lower barriers to entry around new skill acquisition, changing the cybersecurity arms race from one limited by human skill to one potentially only limited by willingness to deploy new AI capabilities the fastest.

However, these risks could be avoided with a different AI architecture. Rather than increase the capabilities of AI in the form of transformer-based language models, we should advance architectures that generate verified outputs with the highest level of safety assurances about their outputs that are provided in other forms of engineering.

Today Formal Verification Is a Critical, Yet Costly Tool to Secure Systems

Formal verification (FV) allows programmers to mathematically prove properties of software for any possible inputs, and is generally considered the gold standard for security and robustness. Formally verified software consists of software, specifications that list goals & constraints, and proofs that the software meets the specification. Examples include:

---

[1] https://www.justice.gov/opa/pr/us-government-disrupts-botnet-peoples-republic-china-used-conceal-hacking-critical

[2] https://www.cnn.com/2024/01/31/politics/china-hacking-infrascture-fbi-director-christopher-wray/index.html Also note: "U.S. officials are concerned the hackers were working to hurt U.S. readiness in case of a Chinese invasion of Taiwan." from https://www.reuters.com/world/us/us-disabled-chinese-hacking-network-targeting-critical-infrastructure-sources-2024-01-29/

[3] https://paperswithcode.com/sota/code-generation-on-humaneval

[4] https://github.blog/2022-09-07-research-quantifying-github-copilots-impact-on-developer-productivity-and-happiness/

[5] https://ee.stanford.edu/dan-boneh-and-team-find-relying-ai-more-likely-make-your-code-buggier

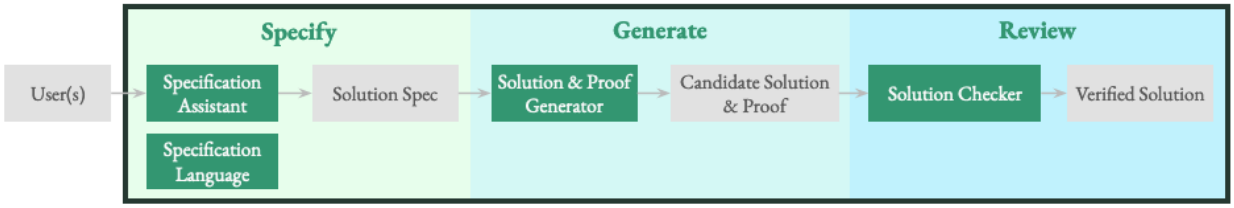
- Formal methods are key in validating high assurance systems, and are generally considered the de facto standard for demonstrating **safety** of flight-critical software systems certified by the FAA[6].
- The DARPA HACMS program yielded a formally verified system that maintained **security** of an autonomous aircraft while under attack from a red team with physical access. One "attacker" even stated "if you fully deployed HACMS technologies ... I may not be able to imagine a way that I could even try to get in"[7].
- Formally verified microkernels (seL4), compilers (CompCert), cryptographic tools (HACL*) and transport libraries (WireGuard, Project Everest), show consumer demand where reliability or efficiency justifies cost.

Verified code in one project was estimated to cost roughly twice that of an analogous unverified system[8]. Creating code, specifications, and proofs may seem inevitably costlier than code alone, but new tools may show this is a fallacy.

AI Could Be Part of the Solution

Instead of building AI-powered tools that reduce the time and thought needed to program, we should build tools that:

- ...generate robust software specifications (and eventually, of general engineering systems) from natural language
- ...help humans understand, compare, improve, or identify edge cases in various specifications
- ...automatically synthesize programs from formal specifications with minimal or no human input required
- ...provide objective proofs (or evidence) that the synthesized programs meet the specifications



These tools would enable systems to be designed, built, and audited with far less specialized expertise, and resulting systems would have verifiable guarantees. Additionally, when dependencies change, formally specified systems could be updated easily (or automatically) by generating updated software from the new dependencies and the old specifications.

This future requires shifting attention to AI architectures that carry more benefits with similar costs and lower risks, rather than advancing risky general capabilities. Ideally, early adopters would find it to be cheap and easy to use formal methods to provide quantitative assurances about the safety of AI outputs, diverting R&D attention from risky general.

What is Atlas Computing Doing

Atlas Computing is supporting development, deployment, and adoption of AI-powered FV tools. Next we will conduct product research, including identifying use cases, generating concept work (e.g. wire frames, requirements documentation, and product roadmaps), and educating potential funders. As a nonprofit, we will openly share this work and help collaborate with anyone working on specification-based software development. Lastly, we want to help ease bottlenecks limiting formal verification, like funding, talent, and job opportunities. These can be respectively addressed by educating policymakers, inspiring students, and coaching researchers to become founders.

Success would look like (1) dramatically growing the FV community (including more funding, trained practitioners, and relevant startups) and (2) incredibly broad adoption of FV tools (ranging from AI tooling for generating high assurance software to dramatically lower time and costs to generate formally verified code).

If you want to use, build, deploy, fund, research, or advocate for AI that makes systems verifiably safer and more secure, please email hello@atlascomputing.org.

---

[6] Rushby, J. (1995). *Formal Methods and their Role in the Certification of Critical Systems* (SRI-CSL-95-1). Computer Science Laboratory, SRI International.
[7] https://www.youtube.com/watch?v=OyqNpn6IpBk and https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5597724/
[8] https://www.youtube.com/watch?v=lRndE7rSXiI