2024 NSF Formal Methods in the Field PI Meeting

Breakout Sessions Summary

Katherine Kosaian Garrett Morris Cesare Tinelli

The University of Iowa

January 2025

The most recent NSF Formal Methods in the Field PI meeting, held on November 12-13, 2024 and hosted by the University of Iowa, allotted significant time for breakout sessions and discussion thereof. In total, there were four breakout sessions on AI/Machine Learning and Systems/Security and two breakout sessions on each of CPS, Hardware, Networks, and Education. Additionally, one group on numerical methods was spontaneously formed, suggesting this as a topic of considerable relevance to the formal methods community. All groups, regardless of subject area, were encouraged to discuss relevant topics related to education. We aim to summarize the discussion of these groups herein, overviewing the written notes provided by the session scribes and identifying key themes.

1 Session Discussions

We turn to an overview of each of the major breakout session topics.

1.1 Artificial Intelligence/Machine Learning

Most of the groups split their discussion by considering both

1. how Artificial Intelligence could be used in Formal Methods and
2. how Formal Methods could be used in Artificial Intelligence.

On the first point, using AI in FM, groups mentioned interest in using AI in synthesis, SAT/SMT, and theorem proving (both for proof generation and for proof repair); here, one challenge is that good benchmark sets are currently not available. Several groups also expressed interest in using AI to generate specifications; one group specifically mentioned the possibility of using AI for analyzing code, but caveated that there was not full agreement on this. There was considerable interest in integrating AI tools into formal methods education, and some interest in using AI to make formal methods easier for students to learn.

On the second point, using FM in AI, much of the discussion seemed to center on challenges and on potential interesting directions for research. Scalability was a common theme, with one group suggesting that targeting surrogate problems could be more tractable and could also help to generate interest/momentum for using formal methods in the AI community, despite the scalability challenges. Other challenges that were discussed include:

- formally *specifying* desired properties of AI systems,
- formally *representing* information for AI, and
- formally *verifying* cyber-physical systems that also include AI components.

Generally, groups were interested in using FM to help ensure that AI systems are functioning appropriately; the focus here is not limited only to correctness but also goes beyond to fairness and robustness. One group raised the idea of using formal methods for probabilistic analysis, focusing not on what the AI could possibly do but rather on what the AI system is most likely to do. Beyond this, there were quite a few ideas suggested for how FM could benefit AI. As a few examples, one group wondered whether formal methods could be used in data mining, another group questioned if (and where!) symbolic reasoning could be used in LLMs, one group suggested developing formal methods techniques specifically for AI focused on programming, and one group suggested using formal methods to verify some of the components within AI systems (e.g., numerical optimization, which ties in well with the numerical methods breakout session).

Dual Problems Notably, it appears that the groups identify some interesting dual problems in ***using FM for AI and using AI for FM***. For example, one group wondered whether AI could be used to help identify false positives in FM use cases; simultaneously, another group wondered if FM could be used to help identify hallucinations in AI models. Similarly, while there was interest in using AI to generate specifications for FM use cases, there was also discussion of how to formally specify desired guarantees for AI models. Additionally, one group noted that brittle proofs can pose a usability challenge for FM in AI, while another group suggested that using FM for AI can help with maintaining these systems over long time horizons. This seems to suggest the key question of ***how AI and formal methods can best interact***. There was clearly interest in this question among the groups. One group expressed a desire to have AI and FM interact more deeply beyond the "common use case" (as they wrote in their meeting notes) where AI is used to obtain some preliminary information before applying FM techniques; this same group highlighted LeanDojo as an interesting example of a "feedback loop" between AI and FM. This ties into another, larger challenge that was expressed by some of the groups; namely, navigating the interaction of formal methods, artificial intelligence, and human users (human-in-the-loop).

Education Groups mentioned that it is difficult to learn both FM and AI. Formal methods, as the groups noted, is not common in undergraduate curricula. Those groups expressed the need to increase formal methods ***outreach to other fields*** and the need for good tool support, with an emphasis on ***tool automation and ease of use*** (or, at the very least, ease of install!). Different ideas of how to integrate formal methods into education were presented; one group suggested integrating them into existing courses in other disciplines; one group suggested having a course on formal methods with (in their words) "AI as application". We observe that another group highlighted that Gagandeep Singh has some good existing course resources on formal methods for AI, which may be of interest here. One group suggested that co-teaching could be particularly valuable for courses that incorporate both formal methods and some other focus. This last group wondered ***whether FMitF would consider funding co-teaching proposals***.

1.2 Systems and Security

Groups identified this as a ***broad and impactful*** area for FM. Many of the discussions focused on challenges and open problems.

Challenges As in the AI/ML session, a frequent theme was the ***challenge of developing correct specifications*** in various domains. ***Tool usability and scalability*** to large, complicated real-world systems were other common themes. Groups discussed how to deal with the complexity of such systems. Reverse engineering was suggested for existing systems (i.e., building a surrogate model of the system and comparing it with the implementation). While one group suggested leveraging compositionality and independently analyzing small pieces of a system, another group (independently) raised the point that sometimes correct components do not work correctly when combined, and highlighted that even specifying how components should combine can be challenging (they cited microservices as an application domain of interest here). One group raised the very interesting question of whether ***having modular proofs can actually help to make software more modular***.

One group discussed what a system is and whether current formal definitions of a system are too limited (e.g., do not include the human element). This same group also discussed what correctness is and how the notion of correctness can be somewhat field-dependent; for example, in business process modeling, there is a notion of an "exception" which is not as strong as a failure. Some participants mentioned that probability and/or fuzzy logic could be helpful here when trying to accurately model (in their words) "fluid definitions". Another group discussed what security is — there are many types of threats, some of which have been successfully addressed by FM and some of which have not. For example, the group mentioned (electric) power-based attacks and noted that even formally determining appropriate guarantees on power is hard.

Open Problems Groups highlighted a wide variety of open problems, including FM for distributed systems (for example, how to handle the failure of a few components in a distributed system, which is relevant for companies like AWS), systems involving LLMs or neural networks, compilers, probabilistic security, privacy guarantees, generating specifications for cloud-native apps, quantum computing, low-level cryptography implementations and, relatedly, hardware security (where working at the appropriate level of abstraction can be challenging), and, finally, regulatory compliance (they suggested Enterprise Governance Risk and Compliance). A couple of groups highlighted that FM could be useful in verifying auto-generated code. As in the AI session, groups suggested using formal methods beyond correctness, citing resilience and performance as other important considerations for FM. There was a recognition that this will require, for example, developing meaningful specifications for suitable notions of performance.

A number of groups discussed the importance of the role of the FMitF program in this area, and some groups raised questions about its role. In particular, there was some discussion of whether/how the program could and should fund efforts to make FM more usable, questions were raised over the differences between the FMitF and SaTC programs. Some participants asked about how to best balance "FM contributions" and "field contributions" in proposals. One group raised the question of whether FMitF could focus more on T2 proposals. Concretely, one group raised the question of how future impactful tools can best be funded — in their words, "How to fund the next Dafny?" While groups note the benefits of industry support and collaboration, they also note potential drawbacks,

e.g., companies may have unrealistic timescales. Support from the FMitF is thus both much needed and much appreciated.

Education Groups identified the importance of interpreting FM education broadly. For example, groups identified the need to train programmers and software engineers and to include FM in software development. As challenges, groups noted that the current usability and scalability of FM is somewhat limited. Some groups suggested focusing on success stories (e.g., in crypto, hardware verification, and compilers; along these lines, one group noted that FM papers have been winning awards at systems/security conferences) in addition to classical examples. At the undergraduate level education, groups suggested integrating formal methods into systems courses, developing an interface that compiles existing FM teaching resources, and developing big online courses. One group suggested that educators would benefit from an agreement in the FM community on what constitutes a "core curriculum" for FM in systems/security. Across the groups, some concrete suggestions of tools to teach in systems courses were: TLA+, Alloy, Forge, Dafny, Verus, and Lean.

1.3 Cyber-Physical Systems

The CPS discussion started on a positive note with a few success stories (such as the verification of ACAS-X) but later suggested that past successes may be a bit of a rarity; rather, ***the success stories may well be a product of current and future research***. The group also suggested that it may help to set lower expectations for the degree of FM that will be immediately meaningful for CPS, and they emphasized the big-picture question of who will make use of or benefit from CPS certification in an applied setting, and what their needs and goals are. Most of the discussion largely centered on ***three major classes of challenges to integrating FM with CPS:***

1. technical challenges,
2. funding challenges, and
3. "workforce development challenges" (in their words).

Technical Challenges Technical challenges for CPS verification that were cited include formally reasoning about nonlinear systems, dealing with uncertainty from a dynamic environment, and gaps between verified models and the corresponding real-world systems.[1]

Funding Challenges There was also discussion of funding challenges for FM research and it seemed to center on ***obtaining funding geared towards making a tangible real-world impact***. Frustration was expressed over the ***challenge of finding funding for "serious long-term tool development"*** (in their words) in the US model; this echoes similar concerns raised by the Systems/Security groups. The CPS group discussed, as an example, the development of the Rocq theorem prover, which was continuously supported by INRIA for more than a decade. However, they noted that US national labs and academia do not always use the same tools, and some national labs (like the DoE) have their own research facilities which are separate from academia. The CPS group also

[1] As models of real-world systems are very complex, often verification cannot handle the full model but only a simplified version, so there is considerable risk that a verified model will not describe the real-world system with enough accuracy.

noted that academic FM research can face a disconnect with industry use and discussed the ***challenge of finding funding to scale FM research into industry needs***. Here, they pointed out that theoretical scaling is different from applied scaling; for example, focusing on improving computational complexity is different from focusing on improving runtime. It was suggested that SBIR and STTR can be useful sources of funding for this; the researchers also appreciated Track 2 of the FMitF program, but suggested that it "does not go far enough" (in their words).

Workforce Development Challenges Regarding industry use of FM, one cited challenge was that engineering has a very different flavor than formal methods. There, ***education and industry tend to be more applied and use tools*** like MATHLAB and Simulink ***which are not always easy for FM to target*** (e.g., Simulink is blackbox and does not have a clear semantics, let alone a formal one). There was consensus that, though the FM topics of interest to engineers will differ by subfield, logic and rigorous specification are a good starting point. Regarding increasing FM education for engineers and industry workers, the CPS group suggested that FMitF Track 3 proposals fund outreach in "unexplored areas" (in their words) and to universities without an existing presence in formal methods coursework, as well as the development of strong online resources.

1.4 Hardware

There were many commonalities between the discussions of the two hardware groups. Both groups cited the importance of ***focusing on emerging and increasingly complicated hardware***, especially heterogeneous hardware, from a formal methods standpoint. As examples, they suggested considering spiking neural network architectures, SmartNICs, GPUs, FPGAs in data centers, in-network computing, and quantum hardware. As one group noted, new hardware needs formal specifications. The groups suggested ***broadening the focus of specifications beyond correctness to also account for security and performance***. One group also had an interest in working towards broadening the focus beyond a very low level of abstraction (where much of the current hardware verification efforts center) to a higher level to enable reasoning about properties of the system as a whole. Both groups seemed to advocate for FM delivering ***correct-by-construction hardware***, which perhaps could be a selling point to industry.

Technical Challenges While acknowledging the historical successes of FM for hardware, both groups identified a number of challenges to wider industry adoption of FM. One group discussed how, ***in industry, performance is paramount***, so industry programmers cannot afford to sacrifice performance in abstractions. To help mitigate this, they suggest a more narrow focus on individual domains, such as computer graphics. This group also mentioned that (in their words) "hardware designers are culturally averse to abstraction".

Both groups identify the need for FM research in hardware security. Notably, both groups identified ***verified security guarantees at the intersection of hardware and software*** to be of particular interest, and both groups cited ***verification for secure enclaves*** as a particular need. One group suggested focusing on "sub-RTL designs in order to anticipate analog side-channel attacks" (in their words). One group suggested that, while performance cannot generally be sacrificed for correctness, verified security is so important that one can afford to sacrifice some performance to attain it.

Education One group discussed challenges related to funding and education. On funding, it was emphasized that projects on applying FM in hardware often need considerable resources; there was a suggestion that ***a combined source of funding from industry and government*** could help to fund work in this space. On education, whether a university offers a formal methods course is hit-or-miss, perhaps because (at present) formal methods courses seem to only be taught by formal methods researchers. To help mitigate this, the formal methods community must ***invest in developing standard resources and course materials that are accessible to faculty outside of FM***.

1.5 Networks

Like the other groups, ***specification*** was a common theme throughout the discussions of the computer networks groups. One group even defined formal methods for networks as centering on specification, verification, and synthesis. There was interest in writing ***specifications to express performance properties***.

Technical Challenges Both groups emphasized the challenge of writing specifications; one group noted that specifications in networks may not express yes/no questions but may instead involve numerical answers, and that correctness might not be deterministic but probabilistic; they suggested (in their words) "[viewing] the network as volumes and flows — where you have continuous quantities approximating discrete ones." One group discussed some strategies for achieving good specifications: integrating specification into industry workflows, perhaps with the aid of tools like Dafny or Verus, and using feedback from the implementation and failures (such as outages) to determine if a specification is weak or flawed. One group noted the challenge of working with preexisting networks as opposed to designing new ones from scratch; they suggested making use of simulations or looking at smaller-scale networks to help overcome this.

Both groups seemed to suggest that it may be beneficial to ***adopt a broad understanding of what network verification encompasses***. One group raised the question of what it would take to achieve a verified network stack. Another group raised the idea of looking not just at the data plane of a network but also at its control plane. One group noted that it would be interesting to ***combine FM with both networks and another area, e.g., distributed systems or cyber-physical systems***. Both groups seemed to acknowledge the existence of some ***overlap between FM for networks and FM for distributed systems/heterogeneous systems***. Similarly to the hardware group, one group noted challenges in using FM for heterogeneous systems/hardware, and suggested that synthesis could be valuable for heterogeneous hardware. The other group pointed out that ***verified guarantees about a network could possibly be leveraged for distributed systems performance***. For example, perhaps optimized protocols could be designed that rely on some established properties of the network.

Other Challenges There was some question over whether there is interest for FM in the networks community, but there was some consensus that FM generally can be valuable for networking and also that ***FMitF funding is valuable for supporting bridges between these two communities***, which are perhaps currently seen as being somewhat separate.[2] One group pointed out the existence of the Internet Engineering Task Force, which works on developing standards (i.e., specifications) for the Internet. There seemed

[2]Interestingly, one of the networks sessions noted that their breakout group had no core people from the field.

to be some consensus that ***more scalability, both of FM and of FM expertise, is needed***.

1.6 Education

The groups largely focused their discussions on ***education at the undergraduate level***. Concern was expressed over the lack of formal methods in the recently updated ACM coursework standards: these standards are quite comprehensive, and yet formal methods is not included in the seventeen outlined "knowledge areas". Both groups suggested ***integrating formal methods into existing courses***, perhaps in part by identifying places where formal methods already exists in some form in those courses. As an example, one group noted that the notion of preconditions and postconditions is already being taught in courses on programming languages or software development.

Specifically, one group advocated a ***bold vision of integrating formal methods into all courses***. They suggest a three-pronged course of action:

1. ***sharing formal methods success stories*** to emphasize the potential for usefulness and scalability,
2. ***sharing useful formal methods tools***, and
3. ***pinpointing how formal methods can integrate into existing courses***.

The second point requires collating, developing, and improving educational resources and materials. For the third point, the group suggests adopting the ***view of formal methods as a skill*** and sharing with colleagues the various ways in which this skill can be exercised (e.g., while coding or designing a system).

The other group heavily emphasized ***developing an inclusive formal methods mindset***. They suggest helping students to come to know formal methods as fun, understandable, and relevant, in part by viewing formal methods as (in their words) "'*tools for thinking*' rather than '*rigorous*'", and by emphasizing how formal methods can help with writing correct code. Concretely, they suggested that focusing on specification, even in the absence of verification, could be valuable in its own right. Interestingly, unlike most of the groups, their ***discussion de-emphasized the role of tools in FM education***. In their words, "tools [...] won't drive adoption outside of experts teaching their particular courses". This group ended their discussion by noting that they hoped for (in their words) "more support from CISE to set the tone for this work".

2 Summary and Themes

Almost all of the breakout sessions noted the ***challenge of writing accurate and expressive specifications***; this appears to be heightened in application domains that are less well-studied, and groups identified many open problems relating to specification. Notably, there was general interest in ***broadening specifications beyond functional correctness to include other notions, such as security, performance, reliability***.

Groups also identified ***usability*** and ***scalability*** as other major challenges for formal methods and for formal methods tools. Generally, groups expressed a ***desire for tools that are more automated (and thus more easily adoptable by industry)***, but noted that ***developing such tools is a long-term endeavor***; on this point, some groups expressed concerns that obtaining funding for tool development over a long time scale is currently challenging.

Despite these challenges, groups expressed a ***desire for FM to have impact in real-world applications and thus an appreciation for the existence and role of FMitF***. Recent and historical success stories sparked some optimism in various discussions, though this optimism also appeared to be tempered with pragmatism given the various technical challenges, as well as perceived challenges in obtaining funding and in increasing the visibility of formal methods.

Most groups highlighted the importance and value of ***increasing formal methods education***; many groups noted that, at least in the United States, formal methods is not a standard part of the undergraduate curriculum, despite the importance of teaching students to reason critically and logically about the code they write and the systems they design. Groups were somewhat mixed on whether formal methods should be taught in dedicated courses or whether it should be incorporated as a module into existing courses on other subjects, with both of the education breakout groups preferring the second option. Multiple groups identified a ***need for standard resources and materials***. Concretely, groups highlighted the importance of ***compiling and curating a centralized set of course materials and accessible FM tools***; one group emphasized the importance of having some resources that are accessible to non-experts. Some groups pointed out the significance of viewing education more broadly and highlighted the importance of ***FM outreach to industry and FM educational resources that are tailored to industry needs***. Some groups also touched on the importance of ***communicating that formal methods is impactful***, e.g. by sharing notable success stories with those outside the FM community. One group in particular encouraged ***sharing the joy of formal methods*** and helping students and non-experts view formal methods as way to think systematically about computational systems.

Acknowledgments

We would like to thank all of the participants in the FMitF PI Meeting for the fruitful discussions; a full list of the participants can be found online at https://cs.uiowa.edu/fmitf24. A special thanks to all those who served as session leaders and to the scribes, who did an excellent job documenting the discussions.

While we aim to summarize and synthesize the written reports from the sessions in this document, we occasionally allow ourselves the liberty of directly quoting from them when their points are worded in a way that we deem particularly impactful (when we do so, we note this with quotation marks).

We are deeply grateful to Allison Rockwell and Matthieu Biger for their administrative and logistical support without which the PI meeting would not have been possible. We also thank our PhD students Sage Binder, Apoorv Ingle, Mudathir Mohamed, and Kartik Sabharwal who provided crucial logistical and technical support during the event. We thank also undergraduate assistants Chloe Ladines and Sydney Libert who provided additional support.

Finally, we would like to thank the FMitF program directors for giving us the opportunity to organize this event. The PI meeting was largely supported by grant 2503109 from the National Science Foundation.