WEBVTT

1
00:00:00.340 --> 00:00:03.600
MEM Producer | Edgar Huertas: Accept the message, and i'm open it up right now.

2
00:01:19.700 --> 00:01:22.370
Chair- Rajeev Alur: Okay. Uh-huh. Hello! Everyone,

3
00:01:23.760 --> 00:01:25.289
Chair- Rajeev Alur: Welcome back

4
00:01:27.450 --> 00:01:28.850
Chair- Rajeev Alur: this stuff

5
00:02:05.000 --> 00:02:21.790
Chair- Rajeev Alur: here. This talks about that. And then, my good, deep sink, Who is the current deputy director for ah Network systems and um. So what distributed systems and effort systems for a key bill for this program, so it's great to have them.

6
00:02:21.800 --> 00:02:30.820
Chair- Rajeev Alur: And then after that we will have reports from the five breakout sessions

7
00:02:31.400 --> 00:02:33.390
Chair- Rajeev Alur: at socks.

8
00:02:33.440 --> 00:02:36.190
Chair- Rajeev Alur: So Michael is over to you.

9
00:02:37.600 --> 00:02:39.029
NSF- Michael Littman: Thanks very much,

10
00:02:43.700 --> 00:02:47.539
NSF- Michael Littman: all right, so I have a five-minute slot, or I should start my timer going

11
00:02:47.620 --> 00:03:03.980
NSF- Michael Littman: five minutes. Seems like a very awkward amount of time. It's like too short for a talk, but too long to just say hello. But I did want to quickly say hello, and then give a very compressed little talk. So Yeah. But thanks, everybody for being here for participating in the meeting. I've been

12
00:03:04.730 --> 00:03:10.249
NSF- Michael Littman: talking with lots of different people, and it's been really exciting it's been. It's been a great time

13
00:03:10.420 --> 00:03:13.170
NSF- Michael Littman: I I, as

14
00:03:13.500 --> 00:03:31.429
NSF- Michael Littman: as Rajek was saying, I am the division director for information and intelligence systems at Nsf. And I started in this role this past summer, rotating from Brown University, but prior to joining Nsf. I was a copi on a formal methods in the Field project joined with the University of Chicago.

15
00:03:31.800 --> 00:03:45.389
NSF- Michael Littman: Personally, I first became interested in formal methods in the context of my research on Reinforcement learning. I was invited to give a talk at Pen, and that's Rajiv Allure's department and folks there schooled me about the kinds of representations being used,

16
00:03:45.400 --> 00:03:51.730
NSF- Michael Littman: and I was intrigued because in part, it seemed like these representations could make reinforcement, learning more effective.

17
00:03:51.750 --> 00:04:01.709
NSF- Michael Littman: In particular. I really liked how logical representations could be used to specify goals in a more direct and meaningful way than reward functions. Oh, I have a slide for that.

18
00:04:01.790 --> 00:04:02.790
Can I switch? Yeah,

19
00:04:02.800 --> 00:04:06.159
NSF- Michael Littman: all right. So I don't know if you can actually see it, though

20
00:04:08.000 --> 00:04:09.100
maybe now,

21
00:04:09.390 --> 00:04:24.870
NSF- Michael Littman: all right. So yeah, So I wanted to be able to specify tasks to an agent. Things like navigate to red, avoiding blue, and to do that in standard reinforcement. Learning you have to decide how valuable are the blue squares, how

22
00:04:24.880 --> 00:04:44.049
NSF- Michael Littman: how much penalty is associated with blue squares. How much value associated with the red squares! If that penalty is too low, you find that agents will do things like walking through the glued against the red, which is not what you wanted. So if you make the penalty for blue higher, in fact, if you make it too high, then the agent doesn't even try to get to read anymore. It just goes and hides.

23
00:04:44.610 --> 00:04:55.070
NSF- Michael Littman: But if you try to specify if you instead, if you specify the task using, say linear temporal logic. You can say things like, eventually, red and always not blue, and get exactly the behavior that you want.

24
00:04:56.120 --> 00:05:00.189
NSF- Michael Littman: So I thought that that seemed like a really nice thing to be thinking about,

25
00:05:00.870 --> 00:05:03.490
NSF- Michael Littman: and as I see it, let's see

26
00:05:03.500 --> 00:05:13.709
NSF- Michael Littman: as I see it, we want to create systems that let people tell machines what they want them to do, and have the machines do those things safely, accurately and reliably

27

00:05:13.810 --> 00:05:29.370
NSF- Michael Littman: accomplishing this goal, in my opinion, requires deep collaboration between machine learning, human computer interaction and formal methods. But specifically, machine learning can learn complex behaviors from example. But the learn behaviors can be brittle, quirky, and hard to confirm.

28
00:05:29.570 --> 00:05:35.399
NSF- Michael Littman: Formal methods can express accurate and meaningful properties, but they can be kind of hard for people to learn how to write

29
00:05:35.910 --> 00:05:40.890
NSF- Michael Littman: tools from human computer interaction can identify and design around those difficulties.

30
00:05:42.510 --> 00:05:50.809
NSF- Michael Littman: So maybe that seems a little abstract. So here's a concrete example that comes out of the fmitf collaboration that I was involved with.

31
00:05:50.820 --> 00:05:55.960
NSF- Michael Littman: So let's say you have a desk lamp in your office, and you'd like the lamp to shut off when it's not being used

32
00:05:55.980 --> 00:06:02.240
NSF- Michael Littman: within a week or two of observing your behavior. A machine learning system can learn to predict when you turn your lights on and off,

33
00:06:02.520 --> 00:06:06.949
NSF- Michael Littman: but, in fact, it could learn many different rules that are consistent with your observed behavior.

34
00:06:07.140 --> 00:06:15.250
NSF- Michael Littman: We found that extracting the rules in a logical form, and then presenting them to the users, let them select the rule that best match their intent.

35
00:06:15.520 --> 00:06:29.790

NSF- Michael Littman: In fact, it was common for them to choose rules that were not the most accurate with respect to the data which is what the machine learning systems wanted to provide, but they better match the sense of what the light should do in the opinion of the person who actually wanted to control them.

36
00:06:30.960 --> 00:06:43.730
NSF- Michael Littman: Now I realize that this example is pretty simple, but I think it suggests a promising way to help make machine learning and Ai more trustworthy by communicating the machine's behavior to the people impacted by it.

37
00:06:43.810 --> 00:07:01.420
NSF- Michael Littman: So, instead of black box, black box, making recommendations about, for example, judicial, sentencing, and parole. We should use representations that are legible. And I see this program at the Mitf program as really pushing the frontier on exactly this kind of strategy. And again, I've been hearing about

38
00:07:01.430 --> 00:07:07.969
NSF- Michael Littman: great work that people are doing across the board, and I think i'm really really pleased with the way that things are going.

39
00:07:08.280 --> 00:07:09.799
NSF- Michael Littman: All right. Thanks very much.

40
00:07:21.700 --> 00:07:24.960
Chair- Rajeev Alur: Learning. Uh, goodbye. You're next.

41
00:07:25.450 --> 00:07:28.199
NSF- Gurdip Singh: Thank you. Thank you. Rajee. Um.

42
00:07:28.340 --> 00:07:56.010
NSF- Gurdip Singh: Good afternoon, and maybe good morning to some of you depending upon where you are. So i'm good Deep Sing. I'm the division director for the computer and network systems. So this is the division which has, uh, you know, the networking program distributed systems program and uh edge to cloud and cyber physical system. All of those programs are posted within within the Cms duration. So clearly, you know, the

43

00:07:56.420 --> 00:08:09.639
NSF- Gurdip Singh: this formal methods in the field program is very relevant to us. Formal methods have a great potential for impact and designing network systems. You know, in systems. We sort of look at

44
00:08:09.650 --> 00:08:18.350
NSF- Gurdip Singh: saying that the services that are offered by a network system should be available anywhere any time without interruption.

45
00:08:18.810 --> 00:08:27.119
NSF- Gurdip Singh: So the thing there is a stat here. We want these services to be available. We just we don't want just any service to be available, our service

46
00:08:27.130 --> 00:08:40.310
NSF- Gurdip Singh: some service to be available. We want the intended service. The thing that you know people want expecting from it which is reliable, trustworthy service, any place, any time without disruption? One hundred and fifty,

47
00:08:40.580 --> 00:09:10.269
NSF- Gurdip Singh: you know. It's almost as you know, we've talked about. You know these network system almost as the ability. If you go into a house, you know, or any place you know you, you look at water, you look at power. These are very reliable things that are available to you. We want iot services network services to be of similar nature, and this can only be made possible by sort of looking at designing them systematically, looking at all the formal methods that we have

48
00:09:10.360 --> 00:09:39.789
NSF- Gurdip Singh: software engineering techniques that we have. You know. We want bring them to bear in terms of designing these. You know our network systems. If you look at cyber physical systems, for instance, these are employed in many safety critical systems, whether it's your autonomous cars, smart building, operating the power grid, and these are critical infrastructure which require that the underlying network system be designed in a in a reliable, trustworthy manner.

49
00:09:39.850 --> 00:10:08.850
NSF- Gurdip Singh: You know, as you move also towards next generation systems. The thing about next iteration system is that they are going to be very, very heterogeneous. They are need to be billions of it. Devices need to be scalable, and these need to be dynamically

reconfigured, as failures are going to be very common. So all of these, again, you know, require this dynamic reconfiguration dealing with therogenity All of these requires all of the weight of the formal methods to bring in, whether you're defining

50
00:10:08.860 --> 00:10:36.589
NSF- Gurdip Singh: proper interfaces. Looking at services that can be composable. All of these, you know, are going to be started, you know. So i'm going to start with sort of taking, You know, researchers from formal methods, field from the systems field coming together and sort of looking at some basic fundamental principles. On how do we build these services which are secured by designs or foundational crack? And that's what you know. This

51
00:10:36.600 --> 00:10:54.250
NSF- Gurdip Singh: This this program was started. Sort of trying to bring these communities together, and we have seen in the past three to four years. You know, a lot of great projects have been funded which has brought the researchers from these different communities together. And you know, and this

52
00:10:54.260 --> 00:10:56.990
the program itself has now has a community on it so,

53
00:10:57.000 --> 00:11:14.510
NSF- Gurdip Singh: and which is great to see. And And so I look forward to. You know the next three to four years of this program whereby we are trying to build these systems, build them to scale. Look at the formal method. Techniques applied to scale, and also also trying to, you know,

54
00:11:14.520 --> 00:11:25.089
NSF- Gurdip Singh: inculcate this in the next generation of graduates that we have, that they also have the

55
00:11:25.100 --> 00:11:28.289
NSF- Gurdip Singh: knowledge of the formal methods to design and deliver these systems.

56
00:11:28.300 --> 00:11:36.609
NSF- Gurdip Singh: So again, thank you all for participating in this Ti meeting, and I look forward to the report that comes on.

57
00:11:36.670 --> 00:11:37.840
NSF- Gurdip Singh: Thanks.

58
00:11:45.480 --> 00:12:01.980
Chair- Rajeev Alur: So now we will have presentations by about the you know the discussion that they had this morning and ah, first it's going to be Ah, Elina and and ask about Ah,

59
00:12:01.990 --> 00:12:03.560
Chair- Rajeev Alur: human-centered computing.

60
00:12:17.370 --> 00:12:18.840
Chair- Rajeev Alur: Okay, Okay,

61
00:12:18.850 --> 00:12:23.109
Chair- Elena Glassman: I can share my screen. But I would prefer to present with with Roz.

62
00:12:23.120 --> 00:12:25.770
Chair- Rajeev Alur: Okay, that that's why it's. So.

63
00:12:30.570 --> 00:12:37.239
Chair- Sanjit A. Seshia: I'm: here. Yeah, i'm ready to go sanjit. How about like I share the screen, and you talk.

64
00:12:37.250 --> 00:12:38.419
Chair- Sanjit A. Seshia: That's great.

65
00:12:38.430 --> 00:12:39.300
Chair- Jerry Zhu: Great.

66
00:12:39.460 --> 00:12:41.720
Chair- Jerry Zhu: So just give me one second

67
00:12:42.320 --> 00:12:44.740

Chair- Jerry Zhu: it's um, but I just

68
00:12:46.460 --> 00:12:47.590
Chair- Jerry Zhu: since

69
00:12:50.650 --> 00:13:06.570
Chair- Sanjit A. Seshia: all right perfect. So so Jerry and I co-chair The discussion session on machine learning we had about twenty to twenty, five participants, and we organized the

70
00:13:07.060 --> 00:13:13.569
Chair- Sanjit A. Seshia: the session around the four questions that we were sent from Nsf. Next slide, Jerry.

71
00:13:15.130 --> 00:13:31.880
Chair- Sanjit A. Seshia: So so. Ah, so here's the first one. Um! What are the success stories for of ah formal methods in machine learning, especially over the last decade. And what has been the impact in terms of advancing research and what's in particular the back-down industry.

72
00:13:31.890 --> 00:13:49.279
Chair- Sanjit A. Seshia: Um. So we try to identify particular topic areas where there's been lots of work on um applying formal methods to machine learning and vice versa. So one of these has been verification of neural networks, particularly feed-forward neural networks.

73
00:13:49.290 --> 00:14:07.920
Chair- Sanjit A. Seshia: And there's been ah a tremendous growth in the literature On this topic over the last five years especially, and it was pointed out that there have been scalability improvements in these tools, which are quite impressive. At least one order of magnitude per year, as observed in the

74
00:14:07.930 --> 00:14:26.849
Chair- Sanjit A. Seshia: the annual Vnn Comp uh competition that runs um going from somewhere from uh, just about three hundred neurons uh in the early benchmarks to to about one hundred million uh uh neurons and parameters in the latest competition. So that's that's been good progress.

75
00:14:26.860 --> 00:14:46.510

Chair- Sanjit A. Seshia: Uh, there's actually going to be a more detailed report coming up. So that's something that we hope to be able to refer to in our uh right up on this um. There have been startups in the space of neural network verification. And yesterday we heard from uh at math works that there's a math box, toolbox, Matlab tool box

76
00:14:46.520 --> 00:14:47.819
Chair- Sanjit A. Seshia: for this as well.

77
00:14:47.990 --> 00:15:06.320
Chair- Sanjit A. Seshia: Um in particular. Ah! Certain ideas, especially those related to abstract interpretation, have not only um had an impact in tools, but they have also ah permeated into the Ml. Community. So there are people working from the Ml. Side who are using these these techniques.

78
00:15:06.330 --> 00:15:11.539
Chair- Sanjit A. Seshia: Um, in In trying to give guarantees about the neural network behavior.

79
00:15:11.880 --> 00:15:27.840
Chair- Sanjit A. Seshia: Um. And then there's also been new methods that that focus on improving applicability of neural network, verification tools to to new problems. We are transformations of the network themselves to make them more aminable to to analysis.

80
00:15:27.870 --> 00:15:41.419
Chair- Sanjit A. Seshia: So that's on the verification of neural networks as components. We also looked at verification of machine learning systems which are, you know, bigger systems like autonomous vehicles that use uh machine learning models as components in them.

81
00:15:41.430 --> 00:15:49.430
Chair- Sanjit A. Seshia: Um. So in this area there's been a lot of progress on simulation-driven verification so-called technologic falsification methods.

82
00:15:49.440 --> 00:16:01.979
Chair- Sanjit A. Seshia: Ah, this is now a scalable technology that's already being adopted by industry Ah, we have open source tools that have been already used in automotive and avionic contexts. Um, even by third-party users of their own.

83

00:16:01.990 --> 00:16:12.509
Chair- Sanjit A. Seshia: There's also been a progress on the other side of using machine learning to improve these methods themselves to improve the search in in temporal logic falsification.

84
00:16:12.520 --> 00:16:23.959
Chair- Sanjit A. Seshia: There's also interesting connections with the work and adversarial stress, testing methods where ideas from formal methods are being combined with ideas from adversarial testing.

85
00:16:24.180 --> 00:16:27.189
Chair- Sanjit A. Seshia: There's also been, if you look further in in

86
00:16:27.200 --> 00:16:42.829
Chair- Sanjit A. Seshia: in terms of full formal guarantees. There's been a nice progress on formal verification methods that are based on hybrid systems verification, such as the very sick tool from pen, and there's a broader competition That's Cps week

87
00:16:42.840 --> 00:16:49.740
Chair- Sanjit A. Seshia: that has been making good progress, and there are several tools already available that participate regularly in these competitions.

88
00:16:49.750 --> 00:16:50.939
Chair- Sanjit A. Seshia: Our next slide.

89
00:16:52.180 --> 00:17:11.350
Chair- Sanjit A. Seshia: Um. There's also been progress on verification of reinforcement learning, particularly in incorporating logical specifications into Ah Rl: Ah! Such as reward design, and so forth. Um, there's been a lot of interest in safe reinforcement learning. Ah, particularly we are the technique that's probably known as shielding.

90
00:17:11.359 --> 00:17:16.950
Chair- Sanjit A. Seshia: However, it has a limitation. That is, it's usually only applicable to known world models.

91
00:17:17.040 --> 00:17:23.699

Chair- Sanjit A. Seshia: And there's also been some progress on verifying Rl: algorithms themselves. That is the the trading algorithms that are used.

92
00:17:24.180 --> 00:17:40.690
Chair- Sanjit A. Seshia: Um! There's quite a bit of interest in on fairness, ah, and analysis of fairness and machine learning from both sides from the machine learning side and the formal methods. Community techniques have been developed for specific definitions of fairness.

93
00:17:40.700 --> 00:17:54.510
Chair- Sanjit A. Seshia: But ah! As many of us know. Ah! There are many flavors of fairness and the ah, and some debate about that. And so that's one of the reasons why uptake of these methods by the Ml. Fairness community seems to be slow.

94
00:17:54.600 --> 00:18:08.629
Chair- Sanjit A. Seshia: Um! There's also been progress from the machine learning community slide and had the signal analysis of fairness. We we we identified this as a as an area where there's been nice intellectual progress, but perhaps not a level of impact that we would like

95
00:18:08.880 --> 00:18:25.160
Chair- Sanjit A. Seshia: um, and then on the side of, but verified or correct by construction, design of machine learning, models, and systems. There are several results on certified robustness for deep learning. We can certainly think of that as a formal methods approach.

96
00:18:25.170 --> 00:18:36.740
Chair- Sanjit A. Seshia: Um! There's been good progress on ah ideas emerging ideas from control with formal methods in machine learning. So things like neural control barrier functions, the up and off control, et cetera.

97
00:18:36.750 --> 00:18:44.470
Chair- Sanjit A. Seshia: And there's been progress on adversarial and counter-example guided training of machine learning models, some of this also being adopted by industry.

98
00:18:44.500 --> 00:18:45.860
Chair- Sanjit A. Seshia: Okay, Next slide.

99
00:18:48.300 --> 00:19:05.149

Chair- Sanjit A. Seshia: Okay, Then we also talked a bit about Ah, the use of machine learning for formal methods. This, of course, is a topic with a long track record. So there's been lots of work on topics like specification, mining and learning like the Invariance, and so on, that have definitely had a big impact in all the methods.

100
00:19:05.160 --> 00:19:19.900
Chair- Sanjit A. Seshia: Ah, more recently there's been a progress on learning temporal logic from Time Series data which has been applied to a range of applications. Lots of work on program synthesis by our machine learning, particularly reinforcement learning.

101
00:19:19.930 --> 00:19:30.870
Chair- Sanjit A. Seshia: Ah, the use of neural guidance, submission, learning, based guidance for solvers. This has been demonstrated for set-solving Qbf solving and empty-solving model counting and other

102
00:19:30.880 --> 00:19:43.989
Chair- Sanjit A. Seshia: uh software applications um. And this is sort of made progress in the academic sense, but it's yet to be seen if it becomes. Ah, something that state of the arts solve was routinely incorporated.

103
00:19:44.000 --> 00:19:59.670
Chair- Sanjit A. Seshia: Um. There's also a lot of interest in machine learning for design, particularly in electronic design automation and for things like interactive theorem proving. And of course, large-language models are now making a splash for both program synthesis and for

104
00:19:59.680 --> 00:20:05.180
Chair- Sanjit A. Seshia: uh, formal reasoning, and that it seems to be another big uh area of opportunity.

105
00:20:05.760 --> 00:20:06.990
Right?

106
00:20:07.000 --> 00:20:21.680
Chair- Sanjit A. Seshia: Yes. So we talked to a went up of the challenges as well. Um! And we've identified several. Um. So you know specification of ah hard to formalize tasks like perception. Ah remains a challenge, and we need better ways to deal with that

107
00:20:21.690 --> 00:20:34.299
Chair- Sanjit A. Seshia: um. Machine learning systems often operate in highly uncertain, unknown environments, and we need to find ways to specify environments to use formal methods. So there's a There's a gap that needs to be bridge there in the right way.

108
00:20:34.340 --> 00:20:48.329
Chair- Sanjit A. Seshia: Ah! Dealing with case scalability in the face of a growing size of deep neural networks. Just as we are able to handle one hundred million. We now have tens of billions of parameters to deal with. So there's always that gap

109
00:20:48.340 --> 00:21:01.899
Chair- Sanjit A. Seshia: We talked about design for verification. You know. How do you design machine learning models that make them more amenable for formal verification. For example, restrict yourself to say things like real use. Ah, so because the tools handle those better.

110
00:21:01.910 --> 00:21:11.869
Chair- Sanjit A. Seshia: Um Compositional reasoning is crucial for formal methods for scale. And yet we don't always have compositional specification because of Ah, for example, the item one on this list

111
00:21:12.150 --> 00:21:24.810
Chair- Sanjit A. Seshia: Um, We talked about how learning theory the community thinks in terms of probabilistic guarantees, like, in fact, learning. And we need to think about how to align probabilistic verification methods better with that

112
00:21:25.450 --> 00:21:38.230
Chair- Sanjit A. Seshia: Ah, machine learning models are but a component of larger systems. And so there was a discussion about how we need more system level techniques to analyze how machine learning performs in context.

113
00:21:38.240 --> 00:21:52.890
Chair- Sanjit A. Seshia: And then we talked about large-language models and the particular challenges that they pose like huge tail the black box nature, because not everybody, has the the Webwall to train such a model. We also talked about how,

114
00:21:52.900 --> 00:22:02.969

Chair- Sanjit A. Seshia: for code synthesis might pose particular challenges because they are producing programs that may have characteristics that may be little different from those that are generated by humans.

115
00:22:03.040 --> 00:22:12.060
Chair- Sanjit A. Seshia: And we talked about many topics, and I just include a citation to a paper that touches upon some of these All right, next slide.

116
00:22:13.450 --> 00:22:28.009
Chair- Sanjit A. Seshia: Um. We also talked about logistical challenges, So the previous slide was more technical challenges on the logistical side. Ah! We talked about how not many people have deep enough knowledge of both areas, particularly machine learning people about formal methods.

117
00:22:28.020 --> 00:22:38.509
Chair- Sanjit A. Seshia: Um. Also the the culture issues. The most machine learning work focuses on achieving performance and less on safety and verification. And you know, how do we change that?

118
00:22:38.520 --> 00:22:55.230
Chair- Sanjit A. Seshia: Um! How do you convince the machine learning community of the value of formal methods in a spanner similar to some, how some other communities, like complete architecture and operating systems, seem to have. Ah started adopting level methods. We need benchmarks that both communities value.

119
00:22:55.240 --> 00:23:05.679
Chair- Sanjit A. Seshia: And then we talked a little bit about large language models, and how to model the processes that people use to design these for particular applications, and what formal methods can offer there,

120
00:23:06.130 --> 00:23:08.830
Chair- Sanjit A. Seshia: all right, next slide.

121
00:23:09.950 --> 00:23:18.160
Chair- Sanjit A. Seshia: So a bunch of recommendations for the future. More cross-fertilization with the learning theory community specifically.

122
00:23:18.350 --> 00:23:33.790
Chair- Sanjit A. Seshia: Ah, maybe three common workshops that that alternate things like that. We need more quantitative and proven metrics that the machine learning community would care about, that The formal methods community becomes aware of and can can design the tools towards

123
00:23:33.800 --> 00:23:47.749
Chair- Sanjit A. Seshia: we'll talk about. You know, more advances in in modular approaches, and I will find more safety, critical applications of machine learning that the community starts to think about similar to how security has become an important driver for various computer systems.

124
00:23:47.800 --> 00:24:03.230
Chair- Sanjit A. Seshia: Um: More work on using formal methods, Methods for machine learning that's used in human centric systems, decision making about people, and we talked about accountability, and how this may be a connection between F. Mitf and the the Das program at Nsf:

125
00:24:03.360 --> 00:24:04.730
Chair- Sanjit A. Seshia: Um.

126
00:24:05.090 --> 00:24:19.309
Chair- Sanjit A. Seshia: Ah, following my third. It was pointed out that it's not just about giving guarantees of safety and ah, insecurity, and things like that. It also enables new paradigms for machine learning like the the interest in ah neuroscience.

127
00:24:19.340 --> 00:24:25.269
Chair- Sanjit A. Seshia: We also talked about Why, how we need better education of the machine learning community

128
00:24:25.750 --> 00:24:27.909
Chair- Sanjit A. Seshia: all right and final slide.

129
00:24:29.050 --> 00:24:41.299

Chair- Sanjit A. Seshia: Um, So how do you demonstrate value? Um, I think one clear way would be to integrate formal methods, tool better with machine learning software. And libraries. There's been some work on that. Um! But we need more.

130
00:24:41.390 --> 00:25:00.279
Chair- Sanjit A. Seshia: Ah, we need demonstrators that really bring out the value of formal methods for Ml. Um, and the community has to think about those. Ah, we need to develop a whole system with you to design and analyze machine learning. I think machine learning people. Ah, very several machine learning people think just in terms of training their models.

131
00:25:00.290 --> 00:25:09.539
Chair- Sanjit A. Seshia: And I think one thing formal methods has to offer is a way to model the context in which the model is used to think about it more rigorously

132
00:25:09.590 --> 00:25:19.530
Chair- Sanjit A. Seshia: we should. It was pointed out that there is some community and machine learning. That's already very enthusiastic about forming methods, and we should leverage that and help that grow.

133
00:25:19.610 --> 00:25:37.439
Chair- Sanjit A. Seshia: Um, We need better ways to educate machine learning folks at about Fm. And maybe organize in a very directed way tutorials at conferences for machine learning people and creating benchmarks of in common interest. And finally it was pointed out that for

134
00:25:37.450 --> 00:25:46.530
Chair- Sanjit A. Seshia: giving guarantees about machine learning, perhaps the incentive structure needs to change about how these are designed, including shifts in the regulatory environment,

135
00:25:46.660 --> 00:25:49.929
Chair- Sanjit A. Seshia: all right, and I think that's it from Jerry and I,

136
00:26:09.890 --> 00:26:13.860
Chair- Rajeev Alur: and maybe the not the power of it. I can answer that.

137
00:26:16.110 --> 00:26:29.789

NSF- Nina Amla: So I think we're planning on making the recordings available. We'll try to make the slides from the industrial panel yesterday available If, if the panelists are willing to share that these slides, I think

138
00:26:29.800 --> 00:26:31.910
NSF- Nina Amla: I would say these are work, product.

139
00:26:32.190 --> 00:26:51.820
NSF- Nina Amla: I'm not sure. The chairs necessarily expect this to be sent out in the current form, but everything that's on the slides will be incorporated into a more polished paper, so that and that information will be available through the paper. Um, Is there anything to add Otherwise, I think,

140
00:26:51.920 --> 00:26:56.780
NSF- Pavithra Prabhakar: Yeah, that's it. I think we'll make the report available once it's ready.

141
00:26:59.790 --> 00:27:01.260
Chair- Rajeev Alur: Yeah, thanks. Yeah,

142
00:27:13.440 --> 00:27:17.580
Chair- Elena Glassman: thanks. I'm just going to share my screen real quick, and we'll get started.

143
00:27:19.200 --> 00:27:21.160
Chair- Elena Glassman: Can everyone see our slides?

144
00:27:21.630 --> 00:27:22.850
Chair- Rajeev Alur: Yes,

145
00:27:23.950 --> 00:27:27.110
Chair- Elena Glassman: all right. Um Ross will will kick us off.

146
00:27:27.300 --> 00:27:45.220

Chair- Ras Bodik: Um, Hi! I hope people can hear me um. So uh, we get the small session number of attendees, but a really good discussion uh it took us a little bit of time to decide if you can slip to the next slide. But

147
00:27:45.230 --> 00:28:01.970
Chair- Ras Bodik: Hcc. Actually is. And and uh, we decided that anything that incorporates the concerns for humans into the design and par writing or evaluation, such as formative and evaluation user studies would be considered human-centered computing

148
00:28:01.980 --> 00:28:17.990
Chair- Ras Bodik: that includes you know a special interactive programming systems for non experts. But they really also developer tools, because programmers are humans, too. And so you would include here things such as academic design of programming languages that are easier to use

149
00:28:18.000 --> 00:28:26.829
Chair- Ras Bodik: in terms of recent successful systems. There are many recent examples that are either commercially available, such as flashf or they are deployed in industry.

150
00:28:26.840 --> 00:28:46.650
Chair- Ras Bodik: They used by engineers or data scientists, or they're influential in the research field and the areas image. These are our range from data processing which could be collection or angling visualization. But then also all the way to classical programming. Where these tools help program developers through your ide ideas,

151
00:28:46.660 --> 00:28:49.639
Chair- Ras Bodik: the successful ideas are

152
00:28:49.650 --> 00:29:14.960
Chair- Ras Bodik: uh viewing artifacts of various kinds as programs. We define semantics that are executable, and therefore you can reason about the verify that test them, synthesize them from data. And so on. These artifacts include cat designs, legal contacts, but also Ml. Models access policies. They can look like rules, but ultimately there are programs and therefore subject to formal methods benefits.

153
00:29:14.970 --> 00:29:24.490

Chair- Ras Bodik: The second idea would be in particular, The humans and designers of tools cannot rely on new interaction models.

154
00:29:24.500 --> 00:29:44.220
Chair- Ras Bodik: For example, in various tools, the idea of manipulating program output has come up. And so you manipulate the output of the program, say you say, cad design, and from that the system reasons backwards and modifies the program that represents that design, or perhaps the input to the program which parameterizes it.

155
00:29:44.230 --> 00:29:59.669
Chair- Ras Bodik: And the third idea was neurosymatic programming, which essentially extends uh synthesis with a neural reasoning, and allows two things to accelerate the search, and it accepts informal specifications. For example,

156
00:30:00.030 --> 00:30:27.749
Chair- Ras Bodik: uh, then, the impact on the various Uh fields. So in advancing formal methods, the incorporation of human concerns led to formative studies which, like to new specification formats which are led to new formal problems, new algorithms, and so on, which hopefully will be generalizable beyond particular interactive tools. And the second was the pressure to generate various more either of counter examples and other.

157
00:30:28.290 --> 00:30:40.289
Chair- Ras Bodik: Ah, the impact in the field are primarily, I would say, the new interaction models that tool designers can rely on, and also auto completion, and I meant to say, output manipulation there.

158
00:30:50.730 --> 00:31:15.199
Chair- Ras Bodik: Um, The impact on industry we mentioned we saw some talks in the industrial session in addition to the policy analyzer data, handling tools and more broadly, Broaderings participation in computing by tools that allow known experts to actually uh do computing work by programming. So you to stargate it for people who are not professional programs,

159
00:31:16.210 --> 00:31:19.590
Chair- Ras Bodik: and I know you can take them next slide or next tool.

160
00:31:20.390 --> 00:31:33.930

Chair- Elena Glassman: Thanks, Roz. So the challenges of actually doing formal methods in this, in humans that are computing is that, first of all, it's not really clear, often to the form of It's not not experts in front of methods. Whether or not

161
00:31:33.940 --> 00:31:57.499
Chair- Elena Glassman: verification or synthesis could be applied um in their situation, or or what the power of it is at all, they maybe realize that they could make use of it. Another challenge is Dsl design So many fields. Don't have good Dsls yet, and someone needs to design them, whether it's someone who's a format of this expert. Or maybe, if we give sufficient tools

162
00:31:57.640 --> 00:32:02.989
Chair- Elena Glassman: to the non experts, once they realized that this power is useful to them.

163
00:32:03.410 --> 00:32:25.550
Chair- Elena Glassman: Another is specification design. So users have this deep domain expertise that shapes what specifications they can provide that are most natural to them that make sense within their own mental model of the problem. But we also have to consider the computational challenges right designing. The Dsl. Can be more like an art than a science where

164
00:32:25.560 --> 00:32:32.689
Chair- Elena Glassman: the algorithm still has to work within certain bounds of space and time.

165
00:32:32.700 --> 00:32:57.080
Chair- Elena Glassman: Finally, the challenge of prototyping and evaluation. So we know, in most human-centered computing activities that if we don't bring the user in as early and often as possible, we may be setting ourselves up to have invested a lot of time and energy in something that is ultimately not particularly useful in the field.

166
00:32:57.260 --> 00:33:14.480
Chair- Elena Glassman: So how can we better involve users in the design process throughout, and what experimental techniques can be best borrowed or invented that are appropriate for evaluating rural methods based approaches specifically

167
00:33:14.490 --> 00:33:17.550
Chair- Elena Glassman: in various Hdc Contexts. The

168

00:33:19.020 --> 00:33:25.620
Chair- Elena Glassman: So our recommendations for the future are to both support a adoption and make prototyping easier.

169
00:33:25.630 --> 00:33:43.029
Chair- Elena Glassman: So field experts should not need to be oral methods, experts to leverage formal methods. Um. But also, if oral methods, experts are doing some deployment, it'd be great to develop better tools and techniques for supporting them in building something for users in new fields.

170
00:33:43.690 --> 00:34:03.390
Chair- Elena Glassman: We also suggest the development of, and and the moment of foundations and tools for semi-automatic design and evaluation of abstractions and dsls the more easily one can play with Dsls and evaluate them perhaps the more quickly we will get to effective ones.

171
00:34:03.560 --> 00:34:32.960
Chair- Elena Glassman: Finally, notion of of Ah, this conversational interactivity that's back and forth between the human and the computer, where the system may provide proposals that need to be presented in such a way that the human can understand, and perhaps refine their own intent and or refine how they're expressing their specifications to gradually refine and revise the the possible programs down to the one that is most appropriate for the situation.

172
00:34:32.969 --> 00:34:42.560
Chair- Elena Glassman: Um, and also hopefully in the process, give the user a more formal understanding of what that program programs would do

173
00:34:42.620 --> 00:34:44.060
Chair- Elena Glassman: by tea. Ross.

174
00:34:48.580 --> 00:34:59.299
Chair- Ras Bodik: The final question was about demonstrating the value, and I don't think there is a better way than building successful systems that I used by people in other fields.

175
00:34:59.310 --> 00:35:13.720

Chair- Ras Bodik: And whether it is data, science, design, machine learning, You know, I, the identification. Those areas where formal methods might be useful is part of the research that we should be doing part of the various meetings and and and summer schools, and such

176
00:35:15.650 --> 00:35:18.889
Chair- Ras Bodik: one of them could be smart homes, and so on

177
00:35:18.900 --> 00:35:32.410
Chair- Ras Bodik: uh how to do it uh it. It also just said that we should meet the users where they are, and understand their workflow so that they build tools that actually fit into how they think about their work, and a fit, together with other tools that they use.

178
00:35:32.420 --> 00:35:41.910
Chair- Ras Bodik: We should also do it by open sourcing libraries for building synthesizers, but available in popular languages and

179
00:35:41.920 --> 00:36:02.409
Chair- Ras Bodik: uh broad, and also the notion of four months of this can be included in it things that are kind of traditionally part of machine learning. So it's also the for programming languages, such as Dsl. Design and implementation, and therefore sort of have a broader scope. That income us uh campus is more experienced and more applications,

180
00:36:02.420 --> 00:36:29.560
Chair- Ras Bodik: and one more slide we have, and to buy an adoption Uh, it would be nice to develop libraries. That territory are easy to use rather than requiring a Phd. In formal methods for parsing. It used to be, you know, decades ago difficult to use, because the algorithm, is very pretty. But now you could pick up parsing in a couple of hours and start using it, even without perhaps taking a compiler class. So can we do that for some

181
00:36:29.570 --> 00:36:48.159
Chair- Ras Bodik: narrow tools in formal methods. Um! Then making the tools executable in web browser so that they are easier to play with, but also the presence in the web browser will make it easier to develop particularly interactive tools that they rely on synthesis, program, demonstration, verification, and such.

182
00:36:48.170 --> 00:37:03.360

Chair- Ras Bodik: And then bootstap collaboration between field experience and formal methods, Experts through summer schools and some sessions where they actually spend some time together. The goal could be to formulate new foremost problems, benchmarks, tasks, and so on.

183
00:37:03.370 --> 00:37:08.320
Chair- Ras Bodik: So that's the broader summary of the session. We are happy to take questions if there are,

184
00:37:34.540 --> 00:37:37.730
Chair- Rajeev Alur: and it feels like you know what it should have.

185
00:37:50.270 --> 00:37:51.379
Chair- Rajeev Alur: So

186
00:37:51.700 --> 00:37:53.490
Chair- Rajeev Alur: with them. Then, just like them.

187
00:37:53.500 --> 00:37:54.290
Yeah,

188
00:37:54.360 --> 00:38:05.899
Chair- Ras Bodik: yeah, I think we wouldn't disagree. I think somebody wrote a code in our dog uh that. Uh, you know, programmers are human, too. And so other researchers, considering how they work.

189
00:38:27.300 --> 00:38:30.240
Chair- Nickolai Zeldovich: Sure, I think, Dan, do you want to share slides, and we can talk

190
00:38:48.230 --> 00:38:50.379
Chair- Deian Stefan: all right. Um! Are we good to go?

191
00:38:50.990 --> 00:39:08.509
Chair- Deian Stefan: So we we didn't present uh, uh, I'll put together a set of slides. But we did our have a somewhat, I think. Okay, organized documents that i'm attempting to shared here.

Let me see if there's an easier way to make it a full screen. But I think it's okay, I says, maybe to um, Okay, so um.

192
00:39:08.520 --> 00:39:25.769
Chair- Deian Stefan: Let's let's get started, so I guess like on the success stories. Um, we. We were thinking pretty broadly. Ah kind of as a systems people. We like the systems to think about systems more generally, and we think there's like quite a few ah places where for myth it's like, really shine. So in in photography, for example, like

193
00:39:25.780 --> 00:39:37.459
Chair- Deian Stefan: Ah, the hackle-star project has kind of been used and is is used in both chrome and firefox it's widely deployed. It's used by basically all of us right now

194
00:39:37.530 --> 00:40:05.209
Chair- Deian Stefan: as as as as a core kind of crypto, primitive uh, but that this is taken on like in other other places as well, So aws is verified. For example, the constant time properties over much of their computer routines. Uh the design of Tls. One point, three itself was actually closely uh done with the formal methods and the academic community more properly right. So a lot of clients or bugs were actually found in the design of Tls. One point three, so that we can, you know, actually ship a thing that is way more secure than

195
00:40:05.220 --> 00:40:11.999
Chair- Deian Stefan: Ah, the the the thing that we did ship before. And then, like you know, there were like dozens and dozens of papers like pointing out bugs that

196
00:40:12.050 --> 00:40:30.619
Chair- Deian Stefan: can be used by attackers. So basically snoop out all our network traffic on the security front. We have things like firefox, and are actually using, like principal formal method techniques like type-based information, flow control to sandbox um different third-party libraries. Um as a general

197
00:40:30.700 --> 00:41:00.150
Chair- Deian Stefan: uh intermediate language uh has itself been designed with formal methods from the start. Right? So it has a formal specification. It has semantics. They are also now mechanically verified. Interpreters are being used as reference interpreters for wasn't itself. So this is used at the micro alliance, which is kind of down during umbrella, composed of large, the industry and a few academic groups. Uh on driving the direction of Boston. Um. There is also like innovation on actually building verified compilers for Web Assembly. Uh,

198
00:41:00.160 --> 00:41:11.919
Chair- Deian Stefan: they're verifiers that are being used today by these companies to actually verify the safety of of what we call compliance code. So there's kind of a lot of things going on in the Security space. And just because of another example here and do

199
00:41:11.930 --> 00:41:37.450
Chair- Deian Stefan: um there, wasn't in front, but uh on the like language and compiler innovation. So there's the uh bug work that actually found tons of bugs and ebpf, and it resulted in new backends for risk. Five that didn't exist before that are like used in in the Linux kernels, like more generally like this community actually has had a tight duration into the Linux kernel, and it's like widely respected. And basically I think any bugs reported have had a huge impact on both

200
00:41:37.460 --> 00:42:07.409
Chair- Deian Stefan: Linux and to everything else that doesn't work on a compiler front. We have things going back to concert, but there's also like new work, right? So there's the work on translation validation, basically being the default and Lbm: So it's finding bugs and and these compilers, and it wasn't filming that much earlier crane lift right there and actually starting to do verified lurings. So we're seeing quite a bit of adoption in in industry, in across the board. So just to like avoid taking way too much time. We've also seen quite a bit on distributed systems. Right?

201
00:42:07.420 --> 00:42:25.669
Chair- Deian Stefan: Tla plus is, I think, that we all hear about that's actually used at places like Vmware and Amazon. It's uh being used to verify Protocols model Checking is kind of like a a a default thing that we're seeing in all these spaces uh in the blockchain world kind of like. If we think of them as distributed systems right like

202
00:42:25.680 --> 00:42:42.719
Chair- Deian Stefan: like all grand, they are actually using machine-check protocols um contract verification in the blockchain world as the default. If it's, not, then we're seeing hundreds and millions of dollars being lost right. So there was a huge motivation for for the verification here. But um, we will kind of

203
00:42:42.800 --> 00:43:09.239
Chair- Deian Stefan: uh looking beyond that. We have uh operating systems embedded systems. And um. Here again we have, like some classic examples like Sel Four. But there's also

like the new work on the um uh the secure Kbm work uh at at Yale, in Columbia. Uh. And there's work like ever parts that is used in the Linux and uh in the windows kernel to just plug in verify parser, so that we're like not getting popped by by by the C. Uh. C partners.

204
00:43:09.460 --> 00:43:38.269
Chair- Deian Stefan: Um, I will let the embedded systems folks talk about Csd embedded systems work, but there's also like a lot of work there. It's like verification based testing and things like that are becoming um outside of kind of the technical direct industry impact. We've also just seen, like a huge rise on communities adopting verification, right? So like the Osdi, the sos of the world. The systems communities like now, like half the tracks, are a certification building verified system. So this is pretty awesome, and we we've seen kind of different people. A lot of

205
00:43:38.280 --> 00:43:40.120
Chair- Deian Stefan: different groups

206
00:43:40.130 --> 00:44:01.390
Chair- Deian Stefan: produce Phd students, and more generally like engineers that have a blend of verification systems. Right? So this is pretty awesome, and also kind of shows that, like maybe, like the fmitf community, doesn't even need to have that requirement like. Have two pis, one systems, one uh verification, because Now we have a whole new generation of of folks that are actually like a blend of the two. So it's kind of

207
00:44:01.400 --> 00:44:08.520
Chair- Deian Stefan: on the programming language side, right like rust is the thing, like all the cool kids are doing it to and rust it. It's

208
00:44:08.710 --> 00:44:32.279
Chair- Deian Stefan: hugely like influenced by the problem methods community. And It's just taking over a lot of like different communities. And this is also kind of exciting, because there's a lot of work on verifying rust. And this means that we can get towards a place where we can actually build verified systems on top of for us, and like, if we verify parts of us itself. Now we have ant and verified systems, and i'll. Lastly, note that there's like a lot of

209
00:44:32.290 --> 00:44:52.070
Chair- Deian Stefan: things happening in industry right so like Amazon, has over two hundred people working on verification. Uh, there's lots of work on verification at Google at Microsoft that much of the different places in the industry. So I guess we're seeing we're maybe like at an

interesting, exciting time in informal Methods committee. But i'll i'll pass that off to to Nicolai for

210
00:44:52.080 --> 00:44:53.130
Chair- Deian Stefan: for the weather,

211
00:44:53.140 --> 00:45:22.290
Chair- Nickolai Zeldovich: sure. So So i'll talk a little bit about the challenges that we see in oral methods, both in industry and in research. Ah, one topic that we all agreed on that was lacking is education and producing enough people that can do verification

212
00:45:22.300 --> 00:45:36.209
Chair- Nickolai Zeldovich: figure out how to do verification in the next round, or build the next version of tools. Ah, but we have a lot of agreement that we need verification to be taught much more foundationally, like algorithms or operating systems. Classes are now, so it would be

213
00:45:36.220 --> 00:46:04.989
Chair- Nickolai Zeldovich: great to have a broadly available courseware. The online classes that people can take or word from.

214
00:46:05.300 --> 00:46:13.639
Chair- Nickolai Zeldovich: So that's one category of challenges is having enough people and having enough education, the other is

215
00:46:13.650 --> 00:46:41.219
Chair- Nickolai Zeldovich: being able to do verification incrementally. There was a broad agreement that there was no way that we could possibly rewrite the world in rust, or call for your favorite framework. So it's important to be able to verify only a part of your kernel or part of your browser, and

216
00:46:41.230 --> 00:46:56.750
Chair- Nickolai Zeldovich: absolutely absolutely not correct. But still there is now being able to do partial verification. So, figuring out how to apply for all methods in this context is an important challenge. How do you deal with existing large systems, where you might not even fully understand

217
00:46:56.760 --> 00:47:04.290
Chair- Nickolai Zeldovich: the behavior or even the behavior of an Api that you're programming against other than some heuristic broad strokes.

218
00:47:04.300 --> 00:47:07.290
Chair- Nickolai Zeldovich: So it seems like an important challenge going forward.

219
00:47:07.300 --> 00:47:14.059
Chair- Nickolai Zeldovich: Another is actually figuring out how to compose verification of large systems that have many challenging aspects.

220
00:47:14.200 --> 00:47:18.030
Chair- Nickolai Zeldovich: So there's heartbrels in their own right that continue to exist like

221
00:47:18.040 --> 00:47:46.799
Chair- Nickolai Zeldovich: it's hard to verify convert code. It's hard to reason about distributed systems. It's hard to reason about confidentiality or weak memory models

222
00:47:46.810 --> 00:47:48.469
how to put them together.

223
00:47:49.240 --> 00:48:14.000
Chair- Nickolai Zeldovich: The related challenge is, How do you actually design for verification? It's often much easier to uh build systems that are verifiable from scratch rather than having to apply verification retroactively. So it'll be useful to understand what advice we could give to make systems verifiable upfront. We had a number of examples of this test, where verification was much more powerful and applicable when designed.

224
00:48:15.380 --> 00:48:30.669
Chair- Nickolai Zeldovich: And then we you have the usual hand ringing that, you know tools are often a challenge. We'd love to have better tools. There's too many different tools. They don't interoperate, et cetera. The documentation is not so great, et cetera. Ah, but that that's our perennials challenge.

225
00:48:32.040 --> 00:48:48.049

Chair- Nickolai Zeldovich: So let me also now talk a little bit about where we see the field going forward, and what the big challenges are. So we try to articulate a couple of different kinds of recommendations or future directions. Some of them are in the form of brand challenges.

226
00:48:48.060 --> 00:49:17.320
Chair- Nickolai Zeldovich: One grand challenge, we see, is proving some really large encompassing systems. So, for example, proving that Whatsapp is correct and secure and will be available despite the tax. So it requires putting together a lot of different kinds of horrible reasoning all the way from the user interface to isolation of different components, networking and distributed systems, encryption, protocols, availability designs, storage, crash, handling all these things. This will require us to

227
00:49:17.330 --> 00:49:37.269
Chair- Nickolai Zeldovich: design components of the system, like the operating system, distributed systems, libraries, et cetera, in a way that are proposable that provide specifications that snap together somehow, and interoperate to allow us to prove this big theory about the overall application, and so on, involves lots of interesting research questions about how to actually build this off.

228
00:49:37.830 --> 00:49:57.790
Chair- Nickolai Zeldovich: Another grand challenge and sort of opportunity is really to see a webassembly as an important building blog that seems to be really on a track to become the standard way to do software fault isolation in the world. But I think there is a big opportunity, and also a grand challenge to apply verification to make this a very robust and

229
00:49:57.800 --> 00:50:17.359
Chair- Nickolai Zeldovich: rounded for formally specified and verified kind of an isolation. And I think there's a lot of good things going for simply that dovetail with verification. But we really need to push to make that happen, because if Webassembly osifies in the next five years without formal verification impact, or

230
00:50:17.370 --> 00:50:20.250
Chair- Nickolai Zeldovich: it might be much harder to make it happen later.

231
00:50:21.150 --> 00:50:38.660
Chair- Nickolai Zeldovich: Another kind of grand challenge is really to come up with precise specs. One particularly acute place where this matters is precisely specifying hardware

interfaces, so that we can build secure systems on top without attacks like spectra and meltdown, and so on, and be

232
00:50:38.790 --> 00:50:42.990
with architects to really pin down what secure hardware should look like. We

233
00:50:43.360 --> 00:51:09.029
Chair- Nickolai Zeldovich: more broadly, we thought it was important for the progress of formal verification to have widely agreed upon specifications for fairly basic things like hardware Isas, but also Api's for binary linking and interaction um systems, interfaces like Wazi for posix like interaction specs for what a file system looks like, what networking primitives look like, and so on.

234
00:51:09.040 --> 00:51:30.210
Chair- Nickolai Zeldovich: There are some prior work that might be good starting points. But this was widely viewed as an important enabling technology, if it existed so good to have, for example, executable specifications that we can test and really good for these specifications to be broadly compatible with a wide range of tools,

235
00:51:31.190 --> 00:51:39.669
Chair- Nickolai Zeldovich: and then more in the aspirational category of how we think systems might be built much better in the future. If formal methods are successful, we thought that

236
00:51:39.970 --> 00:52:05.110
Chair- Nickolai Zeldovich: world of education we really has the potential to allow our systems to be built in a much more extensible way. If we knew that the extensions and modules and changes were safe, or could probably have good performance. So we didn't need to test them or benchmark them, or evaluate carefully, So that seemed like a a powerful enabling technology that might change the way we build systems. And

237
00:52:05.120 --> 00:52:13.109
Chair- Nickolai Zeldovich: oddly, we thought that verification really has the possibility of allowing us to experiment and develop and evolve systems much faster.

238
00:52:13.120 --> 00:52:26.410

Chair- Nickolai Zeldovich: We heard from a couple of people about real industrial systems where the core part of it was held back for a decade, because this was such an important piece of code that

239
00:52:26.420 --> 00:52:36.110
Chair- Nickolai Zeldovich: no one really thought they understood it well enough to have a low enough risk of changing it. There. Wasn't enough confidence that they could change it without bringing down the whole system.

240
00:52:36.410 --> 00:52:54.949
Chair- Nickolai Zeldovich: But we also have had examples where formal verification could come to the rescue and provide a very strong assurance that it was actually okay to make a change to this very four part of the system design and enable performance optimizations that otherwise would have just been rejected for risk reasons.

241
00:52:54.960 --> 00:53:11.739
Chair- Nickolai Zeldovich: Um! Those are those. Those are the big topics. We also thought that a good direction forward would be to address all those earlier challenges we talked about about incremental verification, having better education, materials having better tools, ah, and so on. But that sort of seems to go without saying

242
00:53:13.400 --> 00:53:16.240
Chair- Nickolai Zeldovich: so. Back to that. Um, we'll touch on the last. Yeah.

243
00:53:16.250 --> 00:53:40.419
Chair- Deian Stefan: Um. So uh the the last minute. This is kind of like it is maybe stuff that, if I was already saying is like, How do we actually communicate a bunch of this, and demonstrate the value of our methods to it, to to different communities, or just to to broader kind of uh community outside of even the Cs. Um. So teams things like the blockchain community have done really Well, where verification is the norm. Uh, we. A huge part of this, we think, is because

244
00:53:40.430 --> 00:54:09.159
Chair- Deian Stefan: the cost of not in verification is it's really high right? So this is a case where, like you really want to do this, because, like the alternative is, is maybe not uh such a good outcome, right? That it affects the bottom line. But just on the read there are the communities where this is actually true, but verification hasn't actually take off and taken off as much. And um, they're interesting bits to to, maybe even like, learn from that this community and getting

it like it. Once this thing becomes the norm, then that people just think of it! And if you don't the verification you're that we're right now,

245
00:54:09.310 --> 00:54:17.709
Chair- Deian Stefan: so it may be like a different way to try to get towards this or to try to bridge this gap is to first try to communicate value right? So

246
00:54:17.720 --> 00:54:47.389
Chair- Deian Stefan: one of the things that we heard is developers who don't normally work on a formal methods. They might actually want to, uh learn about formal methods to start reaching out to the problem methods. It seems once they actually have a good understanding of a clear value. So some of those values are things that uh niche like kind of mentioned with, You know that code existing for a decade without being touched, and it wasn't touched because right like it was like on a risk not worth taking. But the verification, folks managed to actually write a spec for it, and then actually improve this piece of

247
00:54:47.400 --> 00:55:15.140
Chair- Deian Stefan: without actually getting up on performance in secure. So we've seen this actually in in multiple cases. There's some cases, for example, like in Firefox, where we would probably be able to write about this. But I think in general having these kinds of examples from industry uh, and trying to like, actually communicate them in either uh industry, experience, papers, or like uh acm uh articles of just written by by people from like kind of positions to communicate effectively to

248
00:55:15.150 --> 00:55:45.010
Chair- Deian Stefan: to to Pm. In in industry. We then kind of trickled out towards our students would be a really big deal towards getting this. Um, There's also just like things that we can do, maybe like without revealing sensitive information from four companies to informal methods. So one one is just trying to bridge a communication gap basically like actually having articles that talk about high profile bugs and how these bugs were found. So these are things that bugs are things that people, I think, have started to actually understand that it's a stuff that we have to deal with, and actually

249
00:55:45.020 --> 00:56:12.810
Chair- Deian Stefan: to explain that, like a bunch of these bugs are actually found with tools developed from the original kind of formal message community. So it's like fuzzing, symbolic execution and things like that. But basically but again, in terms of concepts that people are familiar with and in terms of things that they actually have to deal with on a daily basis, right from browsers to uh, you know, autonomous cars. So basically saying something about the

describing how a car might actually uh care about from methods and why you want that in the software development practice, both of your

250
00:56:12.820 --> 00:56:18.000
Chair- Deian Stefan: a car manufacturer could be a big deal and a thing that the broader audience would care about.

251
00:56:18.010 --> 00:56:46.530
Chair- Deian Stefan: Um. And then I guess the last two bits are again the in industry experience papers, and however, we actually try to like, accommodate for these would be amazing, because this would actually also help us, as like the academics, get back to like the students, or even in our classes, like the interested in in problem. That is, when we're trying to teach about systems and systems like so actually having people saying from these companies say that we do care about these bits, even if it means just how you think about building systems where it would really go right.

252
00:56:46.540 --> 00:57:15.779
Chair- Deian Stefan: Um! And the last bit is really about uh having the meetings or workshops. So I think this uh was a particularly nice uh example for a bunch of us where we actually got to sit up, talk with each other about like the problems that we see. But I more generally actually having meetings where we can involve students industry. Speakers had poly tracks, pretty, awesome and sharing ideas, and in particular showing ideas that I otherwise we wouldn't have it written for from different places, because that could both influence our research direction, but also how we communicate ideas uh back

253
00:57:15.980 --> 00:57:29.890
Chair- Deian Stefan: uh, we are on defense between In-person and not in person. There's pros and cons. I don't think we're going to solve that problem. That seems like a problem that we don't have a good spec for but um it, I I guess we like to meet each other so that that's maybe a good way to get more things going.

254
00:57:31.960 --> 00:57:33.249
Chair- Deian Stefan: That's all we have,

255
00:57:33.860 --> 00:57:36.020
Chair- Deian Stefan: so i'll I'll stop sharing it.

256

00:57:37.410 --> 00:57:39.289
Chair- Rajeev Alur: I thank you, Nicolae.

257
00:57:39.300 --> 00:57:40.290
Chair- Rajeev Alur: Yes.

258
00:57:40.410 --> 00:57:43.450
Chair- Rajeev Alur: Does anyone in the audience have questions,

259
00:58:01.420 --> 00:58:17.309
Chair- Rajeev Alur: but that's you know. I think this is one area where you can be at the point to the success of, because you know, that's

260
00:58:21.290 --> 00:58:24.540
Chair- Rajeev Alur: okay. Let's move to the uh next. Uh,

261
00:58:37.100 --> 00:58:49.009
Chair- Ratul Mahajan: all right. I assume you all can see the slides and hear me. Ali and I are going to talk about what we discussed in network breakout sessions, that

262
00:58:49.020 --> 00:58:57.379
Chair- Ratul Mahajan: sessions is a plural, not not a typo. They were about, I think, twenty, four across tall

263
00:58:57.390 --> 00:59:22.389
Chair- Ratul Mahajan: um, And so we decided to split our discussions into into two groups. Uh just so. Have, you know, Not have too much congestion in in one place. Um, So we're gonna we're gonna split time. But we did try to kind of merge uh the the key sentiments that came out of um both both the sessions, but just the numbers numbers of people should give you an idea how much energy there is, and generally uh applying formal methods to uh networking

264
00:59:22.400 --> 00:59:24.999
Chair- Ratul Mahajan: in there. Uh, to just start with kind of

265
00:59:25.010 --> 00:59:49.369

Chair- Ratul Mahajan: success Stories, I think to kind of maybe dovetailing off of Pallora Jeeves come in there. It is remarkable, frankly, I think the gap between network verification or synthesis becoming like a phrase in academia to its application in industry is just like I've never. I've never seen anything like that take off before from from academia to

266
00:59:49.540 --> 01:00:06.499
Chair- Ratul Mahajan: to to industrial impact for those of you. Now, where essentially, I think, one form of network verification focuses on classical networks like you know, do the control plane verification for network configuration, or do verification for the real pains which is the forwarding state that emerges in the network.

267
01:00:06.510 --> 01:00:27.849
Chair- Ratul Mahajan: That technology now is essentially table states. Ah! To run a large ah tier one network you will not run without it. There's so much kind of um Ah! Awareness of the value it brings, and and result of that also is there's a lot of kind of known risks now to not do that. So all the all the hyper-caters, you know, Google, Microsoft,

268
01:00:27.860 --> 01:00:35.320
Chair- Ratul Mahajan: even kind of Facebook, Amazon, and everybody's kind of published papers and doing and sharing their work with the community around around what they're doing.

269
01:00:35.480 --> 01:00:56.229
Chair- Ratul Mahajan: There's A. There's a few startups. I was involved in a kind of one of them that are taking this technology, and now trying to make it available to not just to tier ones, but to also essentially the the tier two so large e-commerce providers. And and so forth. So this has been, you know, basically you know, unmitigating uh success from from from many ways. You

270
01:00:56.240 --> 01:01:21.690
Chair- Ratul Mahajan: um The second point, I think, just staying within kind of hydroscalers, I think, and some of you again may not know that, uh that three of the four large Cloud providers or Google Google Amazon provide essentially verification under the covers. When you build virtual networks, using their portal uh by teasing buttons or setting some scripts. Uh, you can actually uh prove properties about. You know we can talk about your network,

271
01:01:21.700 --> 01:01:28.840
Chair- Ratul Mahajan: Neha, and they had touched on it yesterday when she talked about network reachability analyzers. But you know,

272
01:01:28.850 --> 01:01:56.730
Chair- Ratul Mahajan: Google has a similar tool, and or it has recently also released, released a tool like that. So these are the companies that are just not just using it internally, but also making it available to to the users as well. So as another success story i'll mention, I think you can say like whether the roots of Sdn are important methods or not, that we can kind of debate back and forth. But one thing to share is like, I think, the formal methods as applied to some of these

273
01:01:56.740 --> 01:02:01.750
Chair- Ratul Mahajan: core problems that emerge in there like consistent updates like you know. How do you update

274
01:02:01.760 --> 01:02:18.359
Chair- Ratul Mahajan: a thousand router network in consistent manner, such that intermediate states can be reasoned about. So that type of thinking came out of, You know, work that is now essentially included in in all Sdn controllers. And then, of course, like

275
01:02:18.460 --> 01:02:25.490
Chair- Ratul Mahajan: networks, big and small, like using Sdn in various ways. So that technology essentially has has a lot of impact there.

276
01:02:25.570 --> 01:02:55.370
Chair- Ratul Mahajan: Um, as another major success story has been essentially you know uh design and verification, and as a modeling language. P. Four. I think this is one where you know a low-level language was being designed, but linguistics and formal methods. People were there, kind of in in ground zero, and that reflects in the design of the language. And you know, A. D four now is fast becoming, not just the programming substrate for a program that would interface cards and switches. But uh,

277
01:02:55.380 --> 01:03:08.760
Chair- Ratul Mahajan: it's also becoming a modeling language, so to speak like Lnb and style in there, so it becomes a basis to kind of formally verify what your network is going to do at a very lowest of levels of packet forwarding in there.

278
01:03:09.440 --> 01:03:37.890

Chair- Ratul Mahajan: Um, The other thing I would say, like, you know, these are kind of we just picked out, you know, a few very concrete kind of big rings, but I would say like from from my perspective, I think the last two are the biggest rains like, you know, I think, the mindset of network engineers and networking researchers because of this collaboration between formal methods people, and I think the mindset has now changed. I think, in your networking. It's. These things are like, you know, um natural concepts to many of us like you know, designing Dsl's

279
01:03:37.900 --> 01:04:07.100
Chair- Ratul Mahajan: uh doing verification, providing some guarantees of certain sorts which you know, a dozen years ago people wouldn't have thought in those terms. Um! A lot of systems people would have thought in those terms. But now it's there, and that gets reflected in the way you kind of think about the way we design the next generation of systems, but it also gets reflected in, you know. Ah, people, we kind of graduate because I think all these industrial successes I was talking about. They wouldn't have happened, you know, had, like, you know, a genre of a a generation of students that i'm gone and made this happen.

280
01:04:07.110 --> 01:04:19.960
Chair- Ratul Mahajan: So I think that's that's also key to remember. I think the change in thinking that has happened from this line of work I would say Nsf. Should get a lot of credit for has led to these things that would essentially will have bigger impact going forward.

281
01:04:21.390 --> 01:04:50.769
Chair- Ratul Mahajan: Um! So let's talk about, I think. Um, so. It's not all you know as anything kind of uh ambitious. It's not always success. I think we did talk about challenges as well. Um, I talked about a lot of them, but I thought we just focus on some numbers uh some small number that kept coming up. Uh, and I think the first one uh, or actually, maybe a lot of these, I think, have have their paradigms in what other speakers have gone. Uh, and i'm doing. The other speakers will come after me

282
01:04:50.780 --> 01:05:02.069
Chair- Ratul Mahajan: lack of data and specification. It's a really acute problem in networking, because a lot of times what you want to verify like properties of traffic properties of configs, the data is not there.

283
01:05:02.080 --> 01:05:19.430
Chair- Ratul Mahajan: Ah! In cases where the data is there, let's say you know, a Tcp implementation in Linux a A formal spec is not there, and so forth. So these are some of the

herdes. Essentially, you know, before you can actually start applying for our methods, you need to kind of ah somehow overcome these challenges to do. Ah, any impactful work

284
01:05:20.370 --> 01:05:48.660
Chair- Ratul Mahajan: beyond data, I think. Um a key, I would say, like, given my experience. Ah, both in industrial research as well as kind. Doing a startup, I would say, like a lot of verification technology now. Ah, the technology itself, the way we kind of think of it. The core is ready. Ah, but really, what holds it back now is like, Can we make it usable in some meaningful way? Ah! Can we develop it in a way that automatically plugs into whatever a network engineer is doing. You know there's no standard

285
01:05:48.670 --> 01:06:00.829
Chair- Ratul Mahajan: Ci, for instance, phone networking where you can just plug this in, and then it will work. So you need to kind of think through what it means for a network engineer to essentially use verification

286
01:06:00.840 --> 01:06:13.719
Chair- Ratul Mahajan: and different people kind of look at it, but it's not something. I think that as a research community gets focused on a lot. But I think but for the next level of impact. That's something to we to get better at

287
01:06:13.780 --> 01:06:15.020
Chair- Ratul Mahajan: um

288
01:06:15.190 --> 01:06:44.980
Chair- Ratul Mahajan: third point here, I think, and um. Others also mentioned this I It is difficult to uh get get students who are trained in both of things uh gone through, you know, and takes, you know, one year, two years to take a networking student, and teach them our formal methods, and take a form that is going to teach them what they're working uh. So I think that that pipeline is not there from undergrads uh level. Uh, we would want you to work in this area. I don't have the right expertise, and that takes time, and that just flows close, everything down as well

289
01:06:44.990 --> 01:06:48.960
Chair- Ratul Mahajan: comes up a challenging a lot of us that that work in this space.

290
01:06:49.760 --> 01:07:05.940

Chair- Ratul Mahajan: Finally, I would say, like, unlike the other two challenges, I think one thing I would say is the way network verification is progress thus far is that the trick, if any of we figured out is how to model our domain in a way that it becomes tractable.

291
01:07:05.950 --> 01:07:35.329
Chair- Ratul Mahajan: Uh using tools. Let's say you know S. And D. Solver, or binding decision, diagram or interactive tour approver our model, checking and so on. But now I think the the next, when it comes to technology development, I think now we are hitting a point where standard techniques and for methods are. We are kind of running out of that, I think, to the next generation of challenges coming up on like if you want to actually more scale than we have today, or we want to validate launch properties in terms of challenges. I think we're getting getting the limits uh limited

292
01:07:35.340 --> 01:07:53.009
Chair- Ratul Mahajan: of that. The final thing is at the bottom of the slide. Here i'd say you can't get a bunch of academics together in a room and just discuss problems without them proposing solutions. So we didn't talk about that. As well take over that, I think, for lack of data and specifications. I think

293
01:07:53.020 --> 01:08:08.960
Chair- Ratul Mahajan: this came up earlier, too, like, you know, we should think about how to have standard benchmarks and competitions. This is where we think like Nsf. Can actually play a lot of coordinate efforts across institutions to help us to help us collectively develop

294
01:08:08.970 --> 01:08:14.969
Chair- Ratul Mahajan: uh either data sources or benchmarks, or maybe just host competitions in there

295
01:08:14.980 --> 01:08:40.779
Chair- Ratul Mahajan: with respect to usability of verification, tools, and everything else. I think we think some some common framework would really help. And there's already, you know, this work in that direction that's happening in Academia, I think, being involved in things like the effort, like Zen which Nick, that I mentioned there and at at Princeton. I think Dave Walker and others did, and be. Where can we actually lift the level of abstraction? And the hope is that if we do that

296
01:08:40.790 --> 01:08:49.550

Chair- Ratul Mahajan: verification becomes easier there, So not everybody has to think too deeply about what verification is doing, and they can maybe think about domain and coding.

297
01:08:49.670 --> 01:09:17.320
Chair- Ratul Mahajan: Um, what's your intent barrier? I think. Um, Some of us have been teaching and creating one of courses on network verification synthesis. One way to kind of amplify, that is, to standardize around like a course that can be taught at other institutions, too, where you know where somebody may not be present or just to kind of build on each other's work and and have a standard kind of you know, if nothing else, a standard language. Talk about these concepts, so we can teach on the next generation of students coming in

298
01:09:17.880 --> 01:09:45.130
Chair- Ratul Mahajan: a final point to think of the scale of quantity in properties. I think we thought it would be good to actually now start looping in, like you know, folks in other domains like optimization, theory or control systems, people where I think, which is where the next, I think next set of problems that folks are failing will come from that, you know essentially not just S. And D solvers or home methods as we think of them. But the other communities doing whose techniques could come to bear in this domain.

299
01:09:45.729 --> 01:09:48.469
Chair- Ratul Mahajan: With that i'll hand it off to

300
01:09:49.689 --> 01:09:51.690
Chair- Aarti Gupta: okay. So touching on

301
01:09:51.700 --> 01:10:18.470
Chair- Aarti Gupta: the next question, which was the future of formal methods in this field aid, or beyond, and identifying some key areas that could benefit most from formal methods. So going from sort of very specific from where we are currency. Ah, for example, in programmable data place where something like before is used. Some of the work that is being done is to develop security solutions,

302
01:10:18.480 --> 01:10:35.589
Chair- Aarti Gupta: but the adversarial and threat models are not are pretty weak right now, so it would be nice to benefit more from the work than insecurities done in formal methods, in order to make those models, as well as the solutions better.

303

01:10:35.600 --> 01:10:37.490
Chair- Aarti Gupta: Another um

304
01:10:37.500 --> 01:11:06.270
Chair- Aarti Gupta: sort of direction in the same context of programmable data plays is that often because of the limited hardware resources in these programmable switches. They have to resort to approximation of data structures like bloom filters, and so on. Ah! Or counts and sketches. But the theoretical guarantees for these data structures are actually much more pessimistic than practical implementations show. So it would be nice to perhaps

305
01:11:06.280 --> 01:11:12.990
Chair- Aarti Gupta: utilize some formal method techniques for specification of performance, for specification of

306
01:11:13.000 --> 01:11:25.449
Chair- Aarti Gupta: accuracy, and then developing solutions to estimate them. Or, again, with some kind of rigorous analysis, to give guarantees not just the theoretical worst case guarantees

307
01:11:25.460 --> 01:11:54.709
Chair- Aarti Gupta: uh moving along. We uh, we've talked about the success of updates network updates, and what happens in practice is in the industry that there's a series of steps taken, and there's kind of like a protocol ban, which is often seen also in other fields like uh, you know, somebody in our session mentioned how it's done in the military, or how it's done in wet lab signs that uh and so there's some check that is done before the

308
01:11:54.720 --> 01:12:16.729
Chair- Aarti Gupta: some check that is done after, but it still feels rather ad hawk, and it's really what the network administrator is doing. So It would be nice to design a language we all love Dsl's and provide a rigorous semantics, so one could actually reason about what these ah update scripts are doing.

309
01:12:17.280 --> 01:12:33.060
Chair- Aarti Gupta: Ah, one of the you know. Again, if you look at the successes, so part a lot of has been done in the context of where they were open systems. But some of the more proprietary stuff it would be nice to. Ah, there's a

310

01:12:33.070 --> 01:13:02.620
Chair- Aarti Gupta: opportunity to also interact more with standards, committees, such as Ietf, because specifications so formal methods is not just verification. It's also specification, and there's huge value in having the specifications be right. They are consistent. They are unambiguous. They are rigorous. So I think that's one of the untapped potential. Some of the people talked about it, and they are working with Ah, for example, looking at the quick

311
01:13:02.630 --> 01:13:09.590
Chair- Aarti Gupta: to call, and so on. But it would be nice to see more work where formula methods could be applied there.

312
01:13:09.600 --> 01:13:20.739
Chair- Aarti Gupta: One of the things that was mentioned a lot in our sessions was, how the next generation of networks are not just going to be about networks alone. They already are

313
01:13:20.750 --> 01:13:34.419
Chair- Aarti Gupta: working with wireless cloud, and there are many cross-cutting concerns, such as virtualization having to worry about cyber attacks. And so it would be nice to basically see

314
01:13:34.430 --> 01:13:49.119
Chair- Aarti Gupta: think about those in relationship to perhaps like the systems Ah, areas where you are talking about, not just the network component, but the bigger system, the bigger. Ah, ah! The thing that it is part of

315
01:13:49.130 --> 01:14:18.249
Chair- Aarti Gupta: likewise uh networks, you know, are an important component in cyber physical systems, in applications such as autonomous driving. So it would be really nice to reach out to some of the work going on in these adjacent fields, because that's where the next step that works about in terms of the advances we felt are needed in F. In order to address some of these challenges. Certainly, in order to do a quantitative, reasoning,

316
01:14:18.750 --> 01:14:22.870
Chair- Aarti Gupta: probabilistic reasoning, some kind of stochastic modeling,

317
01:14:23.260 --> 01:14:44.610

Chair- Aarti Gupta: especially when we go to cyber physical systems, then adoption of some control theoretic techniques, and also for performance and estimation and other things. Other reasons it would be useful to ah borrow from the rich Ah! Topics and a wide variety of techniques available in optimizations

318
01:14:45.000 --> 01:14:46.469
Chair- Aarti Gupta: moving on

319
01:14:47.540 --> 01:14:49.110
Chair- Aarti Gupta: the next slide.

320
01:14:49.260 --> 01:15:06.709
Chair- Aarti Gupta: So, in terms of demonstrating value. How can we communicate and demonstrate the value within and beyond traditional Cs audiences so ruther mentioned that the big success has been certainly with hyper scalars. But I think

321
01:15:06.720 --> 01:15:23.270
Chair- Aarti Gupta: once we get these formal ah method, tools, or network verification tools to become more usable, I think. Then there's an opportunity to also engage more broadly, not only with hyper-scalers, but but with other industries;

322
01:15:23.280 --> 01:15:51.440
Chair- Aarti Gupta: and we would again, like it, has been mentioned by other chairs we'd like to communicate to communities that use networks as a component, because ah, certainly in the ah conferences, like Nsdi sitcom. Ah, it's very well understood the value that formal methods bring to network gratification and network design. But maybe outside of the community, people who use networks, but may not be aware that the networks

323
01:15:51.450 --> 01:16:13.600
Chair- Aarti Gupta: a community is already adopting and using formal methods. It would be nice to communicate that to other communities as well. And one way we could do that Ah is to publish accessible case studies again, maybe in Csm. Articles or other articles that have more broad audience, not just our conference proceedings.

324
01:16:13.610 --> 01:16:18.500
Chair- Aarti Gupta: One other suggestion that came up is that

325
01:16:18.510 --> 01:16:47.989
Chair- Aarti Gupta: I hope it was clear from a lot of the industry Session talks yesterday as well. That formal methods in networking has had a huge impact in industry practice, especially in saving, perhaps I mean Yes, whenever there's an outage there's a big news flash that there was an outed, and it costs so many millions of billions of dollars. But it would be, you know, nobody publicizes when it doesn't happen, or something got safe, so it would be good to figure out how to measure that kind

326
01:16:48.000 --> 01:17:04.290
Chair- Aarti Gupta: of impact, and whether the amount that is being saved in the industry is tenx, one hundred x thousand nets. What's the order of magnitude in terms of what perhaps Nsf. And other government agencies are putting into uh such programs.

327
01:17:04.300 --> 01:17:14.009
Chair- Aarti Gupta: It would be really nice to have handy something to quote for the cost, say to the industry, for example, for us to write future proposals in this space,

328
01:17:14.510 --> 01:17:43.210
Chair- Aarti Gupta: and then, you know, going beyond networks. Some of the other applications and domains that we discussed were, of course, you know, following the money or following domains where there's a lot of repetitive procedures, things like healthcare, pharmaceuticals algorithmic training. All of these seem again, you know, hugely important from the point of your code financial. Ah, gain as well as you know safety critical.

329
01:17:43.220 --> 01:17:45.269
Chair- Aarti Gupta: So

330
01:17:45.940 --> 01:18:02.380
Chair- Aarti Gupta: it would be nice to see more formal methods. In some of these there are, you know, isolated things, but in terms of ah, a bigger and wider ah effort in these domains. Uh, we felt it would be nice to see that.

331
01:18:02.390 --> 01:18:06.939
Chair- Aarti Gupta: And let me just add a very quick note at the very end, the last slide

332
01:18:07.000 --> 01:18:35.340

Chair- Aarti Gupta: uh one of the other things. Uh, you know we all have this fi meeting, but it was great, and thanks to Nsf. And the organizers, but it also felt like we had very little time for interaction with other Pis, especially in the lightning talks. Uh, it was great to see what those were, but there were there was hardly any time for questions, and also it would be nice to exchange ideas across the different fields. We are getting some of that in this last session, which is great.

333
01:18:35.350 --> 01:18:58.490
Chair- Aarti Gupta: Um, but would be nice to maybe have more time to chat with some of the Pis, especially for the networks people in the Cbs domain, or people who are looking at autonomous driving intelligence systems, and so on. So, thanks again to everyone. But we'd love to find. Figure out a good way of interacting more with the Pis and across our.

334
01:18:58.500 --> 01:18:59.639
Chair- Aarti Gupta: Thank you,

335
01:19:03.940 --> 01:19:06.840
Chair- Rajeev Alur: Thanks. At the end of it

336
01:19:30.110 --> 01:19:34.900
Chair- Rajeev Alur: so. Yet the last talk is by,

337
01:19:34.910 --> 01:19:37.880
Chair- Rajeev Alur: and make me, I think, make me a

338
01:19:54.890 --> 01:20:05.459
Chair- Sayan Mitra: all right, assuming you can hear me. Okay. So yes, this was the session on embedded systems,

339
01:20:05.470 --> 01:20:22.389
Chair- Sayan Mitra: a moderated in the discussions. So this group was a bit more diverse. Um. So we had members from the cyber physical systems community from the high performance, computing community autonomous systems as well as Ah New Michael Computation.

340
01:20:22.400 --> 01:20:38.489

Chair- Sayan Mitra: So, just to give you a sense of the members who were participating in this discussion. Let me show you the list of speakers from yesterday. During the discussion we had several other members.

341
01:20:38.500 --> 01:20:41.659
Chair- Sayan Mitra: The group was about twenty people on the average.

342
01:20:43.050 --> 01:20:53.680
Chair- Sayan Mitra: Yeah. So we started out discussing the successes of formal methods in embedded systems, and there are quite a few

343
01:20:53.690 --> 01:21:07.559
Chair- Sayan Mitra: the successes to talk about. We note a few over here. So, for example, twenty years ago, there was very little awareness of formal methods at mathematics which is

344
01:21:07.570 --> 01:21:32.910
Chair- Sayan Mitra: develop. This is a very popular network simuling tool. And now there is, as you heard from shares, talk yesterday. They have several groups using formal methods and notably specification guided. Testing is becoming a standard practice. Ah! There is also the simulink ah design verifier Ah! Which is coming directly from this formal methods community.

345
01:21:33.210 --> 01:21:44.339
Chair- Sayan Mitra: And similarly, in automotive companies like Ford and Toyota. A decade ago there was very little formal methods being used,

346
01:21:44.350 --> 01:22:01.289
Chair- Sayan Mitra: and now methods are getting transitioned into the industry. There is a code generation from models. There is verification of controllers, and also, of course, alongside traditional types of testing.

347
01:22:01.300 --> 01:22:03.929
Chair- Sayan Mitra: Uh, one of the

348
01:22:03.950 --> 01:22:20.250

Chair- Sayan Mitra: the things that came out of this type of collaboration was this: our brain benchmark from several researchers at Toyota, which actually also had a big impact on academic research, informal verification of control. Systems.

349
01:22:21.010 --> 01:22:39.160
Chair- Sayan Mitra: The other area that was mentioned is in high-performance computing. For example, polynomial analysis has now enabled extraction of unseen parallelism in loops, and these techniques are becoming a part of the standard compiler Architecture, for example, through Llvm

350
01:22:39.640 --> 01:22:58.569
Chair- Sayan Mitra: now. Ah, the other Ah, broad point that came out of this discussion was that um type systems have become a big success story, and we didn't have specific companies or start-ups here. But the broad point was that

351
01:22:58.580 --> 01:23:18.049
Chair- Sayan Mitra: um web-scale system development has shifted the the appetite of developers in using formal methods, not not just for bug finding, but also during the development process, to to avoid a whole family of bugs.

352
01:23:18.060 --> 01:23:29.350
Chair- Sayan Mitra: So things like typescript are becoming very popular, and similar ideas are making their way into other programming languages like Java and Russ.

353
01:23:30.590 --> 01:23:48.790
Chair- Sayan Mitra: Now, the other thing that was interesting to note, and this was also going to come up later on in my presentation is that traditionally formal verification has been used as a way of

354
01:23:48.800 --> 01:24:02.889
Chair- Sayan Mitra: gaining correctness or finding bugs. But there is another emerging usage of formal techniques which is worth paying attention to, and that is in managing complexity of systems

355
01:24:02.900 --> 01:24:20.220

Chair- Sayan Mitra: as a software, and hardware gets developed in inappropriate and getting deployed faster, and they are being built by large teams and changing environments, managing the interfaces and the components is a huge challenge,

356
01:24:20.230 --> 01:24:28.520
Chair- Sayan Mitra: and in many cases this is a new opportunity for applying or developing new form of,

357
01:24:28.530 --> 01:24:48.720
Chair- Sayan Mitra: and this is an interesting contrast with the traditional usage of formal methods where you know a very large development cycle, and you have to worry about failure of a single component, and a lot of energy goes into that type of verification that traditional models say Nasa or the chip design industry.

358
01:24:50.130 --> 01:25:03.009
Chair- Sayan Mitra: And of course we also talked about successes in transitioning Phd students into the industry. And this is not just the matter of

359
01:25:03.020 --> 01:25:16.099
Chair- Sayan Mitra: ah workforce development. But for truly transitioning formal technology in the industry. This is important, because usually selling a formal verification tool is very hard.

360
01:25:16.110 --> 01:25:28.909
Chair- Sayan Mitra: And so the knowledge front so really has to happen through people and Bhd students. When they go off to work in a certain company they often cede new ideas or groups

361
01:25:28.920 --> 01:25:37.269
Chair- Sayan Mitra: for developing a formal technique. So this has been a way of transitioning technology into the industry as well.

362
01:25:38.380 --> 01:25:53.630
Chair- Sayan Mitra: Ok. So, while on the topic of education, some of us in this discussion group, very, very pleased to find out about this software foundations project. I didn't know about it, And this is like an excellent resource

363

01:25:53.640 --> 01:26:11.309
Chair- Sayan Mitra: for education. So I put the link here. You can find out more about it If you, Haven't looked at this resource for learning about formal foundations or programming languages and verification, and there's a lot of ongoing development around this project.

364
01:26:13.250 --> 01:26:29.589
Chair- Sayan Mitra: Okay, So those were some of the success stories. And then we talked about challenges, and broadly we talked about two classes of challenges, and the first one is the the usual one. So lack of benchmark,

365
01:26:29.600 --> 01:26:32.590
Chair- Sayan Mitra: particularly for embedded systems

366
01:26:32.600 --> 01:26:52.909
Chair- Sayan Mitra: and control systems. It's a difficult to find benchmarks, real World benchmarks. I mentioned the power train control system in the earliest slide as a as a real exception, and this has a straight jacketed research. Quite a bit. So academics work on tortoise problems

367
01:26:52.920 --> 01:26:56.020
Chair- Sayan Mitra: and difficult to scale these problems.

368
01:26:56.030 --> 01:27:24.669
Chair- Sayan Mitra: And, on the other hand, it's difficult for industrial researchers to share real world code or real-world ah controllers for for guiding the research, and particularly this was noted in the case of embedded system, security. And while there are some small scale benchmarks a large scale. So sea level benchmarks are absolutely seen as a barrier to advancing research.

369
01:27:25.400 --> 01:27:42.370
Chair- Sayan Mitra: We also talked about ways of overcoming this challenge, and one of the idea here was to actually create benchmarks based on educational avenues that are available. For example,

370
01:27:42.380 --> 01:27:57.639

Chair- Sayan Mitra: hardware, designing, hardware design is becoming more and more open source driven risk by. There are tools now for going all the way from very log to layout and implementation of

371
01:27:57.650 --> 01:28:13.739
Chair- Sayan Mitra: microprocessors almost entirely through open source tools, and this provides an avenue to create these pipelines, perhaps through classroom activities and research projects

372
01:28:13.750 --> 01:28:27.899
Chair- Sayan Mitra: which will then translate into actual circuits and designs for advancing verification. Research. And so it was noted that there were several open source frameworks for doing these types of things.

373
01:28:28.700 --> 01:28:33.410
Chair- Sayan Mitra: Another similar endeavor is this autonomous racing competition?

374
01:28:33.440 --> 01:28:49.280
Chair- Sayan Mitra: Ah! Whereby the project is gaining inputs from the community in the form of controller code. And then we can use this controller code to as target for formal Verification research.

375
01:28:49.770 --> 01:29:08.610
Chair- Sayan Mitra: So that's the usual sort of complain we have in finding benchmarks. Then the discussion about challenges became quite interesting, and we talked about this issue of fragmentation of languages. And

376
01:29:08.620 --> 01:29:30.870
Chair- Sayan Mitra: so the issue here is that there are many parallel efforts in different formalisms and different mathematical languages which are potentially overlapping and duplicating some of the efforts, for example, formalization of certain types of arithmetic, kit and pbs and f-star so forth.

377
01:29:30.880 --> 01:29:36.559
Chair- Sayan Mitra: A potentially, you know, duplicating some effort

378

01:29:37.000 --> 01:29:44.939
Chair- Sayan Mitra: it is extremely difficult to port proofs across these different formalism,

379
01:29:45.170 --> 01:29:56.299
Chair- Sayan Mitra: and that's staying within the world of interactive. You're improving. Now, if you try to go across the peer improvers and model checkers, this challenge becomes even harder.

380
01:29:56.730 --> 01:30:16.410
Chair- Sayan Mitra: So this is a serious issue which slows down research. Perhaps the smt world is less affected by it, thanks to all the work that has gone into creating standardized libraries and interfaces. But this is a serious challenge in pushing forward

381
01:30:16.420 --> 01:30:18.799
Chair- Sayan Mitra: basic ideas, informal methods.

382
01:30:19.140 --> 01:30:32.719
Chair- Sayan Mitra: A similar problem was noted also at the level of languages. For example, plcs in control systems use certain languages which are hard to formalize and connect with existing formal.

383
01:30:33.780 --> 01:30:50.979
Chair- Sayan Mitra: So again, this fragmentation problem we discussed some challenges as some of the solutions to this problem, and the one that really stood out was this idea of using critical mass, leveraging critical mass

384
01:30:50.990 --> 01:31:04.739
Chair- Sayan Mitra: uh, perhaps sometimes at the expense of expressive power, or some of the other things that we would be nice to have for purely from the formal verification point of view.

385
01:31:04.820 --> 01:31:23.479
Chair- Sayan Mitra: So A. C in point. Here is widespread adoption of C. For example, it has many limitations, but people still use it because there are so many popular and good libraries available. Similarly, in the world of your improving

386
01:31:23.490 --> 01:31:38.759

Chair- Sayan Mitra: um clock has a large user base at this point, and many tools supporting libraries, and so it has become the dominant framework for developing theorem upper-based technologies.

387
01:31:38.880 --> 01:31:54.779
Chair- Sayan Mitra: So this has worked for some of the proof assistance, and it is because there has been this long-standing support from government agencies For example, in the case of

388
01:31:54.790 --> 01:32:05.539
Chair- Sayan Mitra: this has been possible because of thirty plus years of support of the French Government. It was noted that there was a similar team trying to do

389
01:32:05.550 --> 01:32:16.790
Chair- Sayan Mitra: a common framework based on neutral. The theor improving system decades ago in the Us. But did not gain as much funding. So that was limiting.

390
01:32:17.360 --> 01:32:26.799
Chair- Sayan Mitra: And so further, talking about this idea of building broad ecosystems for software development and languages.

391
01:32:26.860 --> 01:32:46.520
Chair- Sayan Mitra: Ah, someone pointed out this Nsf Pros program, which is precisely for supporting software development projects. But you know, to build a large ecosystem of interacting tools three to four years may not be enough. So here we have some

392
01:32:46.530 --> 01:33:05.809
Chair- Sayan Mitra: um social social challenges like how to support a broad-based tool development which itself take time and money, but also takes ah users to to gain this type of critical mass that is necessary to bridge across these different silos within form of verification.

393
01:33:06.760 --> 01:33:12.059
Chair- Sayan Mitra: So the finally, with on this topic of fragmentation, it was noted that sometimes

394
01:33:12.070 --> 01:33:35.110

Chair- Sayan Mitra: agmentation may not be avoidable, and maybe even necessary. For example, if you want to develop techniques which are specific to a domain or exploit the structures that are seen in a particular domain, then loving and domain-specific technology may be useful for verification,

395
01:33:35.120 --> 01:33:48.470
Chair- Sayan Mitra: and so one has to balance this trade off between being super domain specific and versus siloed and disconnected from other other technologies.

396
01:33:50.030 --> 01:34:01.209
Chair- Sayan Mitra: Okay recommendations for the future. We agreed that keeping the all proposal broad, helped

397
01:34:01.290 --> 01:34:08.890
Chair- Sayan Mitra: help to extend the audience beyond the traditional form formal methods. Community

398
01:34:09.040 --> 01:34:28.669
Chair- Sayan Mitra: Um. The other recommendation was to move towards code level verification, particular in the context of cyber physical systems. We saw a big successes of Ah, deploying verification at the code level and the Ci pipelines in the binary docks yesterday.

399
01:34:29.050 --> 01:34:53.939
Chair- Sayan Mitra: Um, so. And of course, even in the case of control systems, Ultimately, what is getting deployed is code, and sometimes it's hard for end users and developers to convert their code to models, and they may not have the time and the bandwidth to do that. So techniques that can work directly on the code potentially have bigger users.

400
01:34:54.010 --> 01:34:55.230
Chair- Sayan Mitra: Um!

401
01:34:55.340 --> 01:35:03.210
Chair- Sayan Mitra: Then, bridging this gap between code and model. It may not be just a one-step solution,

402
01:35:03.220 --> 01:35:18.180

Chair- Sayan Mitra: but other interfaces and layers may be necessary for example, generating code from some higher-level languages to lower level languages to things that are closer to floating point specifications.

403
01:35:18.190 --> 01:35:26.790
Chair- Sayan Mitra: So there might be some work needed to close these gaps, using, say, functional modeling languages.

404
01:35:28.330 --> 01:35:40.599
Chair- Sayan Mitra: The next thing we talked about was going back to this idea of managing complexity, of fast-changing and large-scale systems

405
01:35:40.650 --> 01:35:48.179
Chair- Sayan Mitra: here the problem is different from the traditional scalability issues

406
01:35:48.190 --> 01:36:00.860
Chair- Sayan Mitra: that formal verification typically addresses so developing compositional techniques and techniques for incremental verification, exploiting monotonicity in models

407
01:36:00.880 --> 01:36:12.579
Chair- Sayan Mitra: sort of focusing energy on these types of problems can potentially help make a bigger impact in the field.

408
01:36:13.620 --> 01:36:14.840
Chair- Sayan Mitra: Um,

409
01:36:15.150 --> 01:36:25.159
Chair- Sayan Mitra: and then support for community building tools. We already talked about it in the previous slide as a way of bridging these different silos.

410
01:36:25.320 --> 01:36:44.040
Chair- Sayan Mitra: Um. And one way that we can gain more traction in newer fields is to talk about formal methods as basically better testing or to say, Oh, if your system is too hard to simulate,

411

01:36:44.050 --> 01:36:49.739
Chair- Sayan Mitra: then here is a method that you can use to to cover more cases.

412
01:36:52.280 --> 01:37:00.650
Chair- Sayan Mitra: Ok. So, finally, talking about the different ways to demonstrate value of formal methods,

413
01:37:00.680 --> 01:37:09.960
Chair- Sayan Mitra: we started off with a discussion on improving awareness of the capabilities of formal methods. Yes, it was recognized that

414
01:37:10.040 --> 01:37:27.010
Chair- Sayan Mitra: although smt solvers and saxophones have become very, very powerful, many of our undergraduates still do not know about these tools. It is unfortunate. It was noted that in this Hpc community, for example, where formal methods is making interesting impact,

415
01:37:27.020 --> 01:37:45.729
Chair- Sayan Mitra: one of the biggest meetings recently there was only one workshop of one workshop, maybe, which we touched upon formal methods. So there are wide swains of computer science, even within which the capabilities and the latest developments of formal methods are not known, so we could do a better job of

416
01:37:45.740 --> 01:37:51.249
Chair- Sayan Mitra: outreach and connecting with the untapped parts of this market.

417
01:37:52.010 --> 01:37:53.240
Chair- Sayan Mitra: Um.

418
01:37:53.960 --> 01:38:18.490
Chair- Sayan Mitra: Some in their group were of the view that you know each of these fields have to go through their own intel emptive moments, so to speak, to realize that they need formal that another variation of that is these disaster stories, right? So once some ah really bad thing happens because of the software bug, then the community wakes up, and Ah adopt some of these techniques,

419
01:38:18.500 --> 01:38:21.999

Chair- Sayan Mitra: and we have hard ecos of this from the other talks as well today.

420
01:38:22.670 --> 01:38:34.339
Chair- Sayan Mitra: But then we can be more proactive and develop tools that are easy to use without necessarily understanding all the underpinnings of formal methods and the theory.

421
01:38:34.350 --> 01:38:44.760
Chair- Sayan Mitra: So, for example, from the machine learning community tensorflow and automatic sensation has become so easy that undergrads can even use it without

422
01:38:44.770 --> 01:38:59.250
Chair- Sayan Mitra: totally understanding the details of Gpu programming and the math behind, always so pushing to make formal methods tools usable is definitely a way forward.

423
01:39:00.020 --> 01:39:14.759
Chair- Sayan Mitra: Um! And then again going back to this idea that beyond just correctness and finding Bugs formal methods as a way of managing complexity might have a broader appeal.

424
01:39:14.770 --> 01:39:30.129
Chair- Sayan Mitra: So work on api's and specifications for interfaces of different large-scale models reasoning about directness at that level, but as a way to sort of write the court tales of

425
01:39:30.140 --> 01:39:35.199
Chair- Sayan Mitra: new capabilities that these techniques can provide.

426
01:39:37.370 --> 01:40:02.869
Chair- Sayan Mitra: And on the same topic. You know, new capabilities can be in the form of synthesis and automation. So, and under the hood we might. This community might recognize that ultimately synthesis and verification might be very closely related. But it might be much easier to sell these as new capabilities than ah desktop findings.

427
01:40:04.160 --> 01:40:23.120
Chair- Sayan Mitra: Ah! And finally Ah goes without saying that sustain funding for ah formal methods, for example, through programs like formal methods and field, can help convince Ah departments where currently there is no presence of formal methods

428
01:40:23.130 --> 01:40:37.300
Chair- Sayan Mitra: to recruit faculty in this area, and also to create strategic workshops which can help connect the gap or bridge to newer fields where formal methods can have.

429
01:40:37.790 --> 01:40:42.699
Chair- Sayan Mitra: So I think those are the main points. Happy to take questions if there are any.

430
01:40:45.870 --> 01:40:48.029
Chair- Rajeev Alur: Well, thank you, Shannon:

431
01:40:48.830 --> 01:40:51.590
Chair- Rajeev Alur: Yeah, it's kind of one last question,

432
01:40:51.910 --> 01:40:53.510
Chair- Rajeev Alur: Yeah, please,

433
01:40:54.420 --> 01:40:55.820
Chair- Rajeev Alur: for them,

434
01:40:56.680 --> 01:40:58.080
Chair- Rajeev Alur: since two thousand and eight

435
01:41:08.850 --> 01:41:10.059
Chair- Rajeev Alur: and

436
01:41:10.110 --> 01:41:13.800
Chair- Rajeev Alur: some more standard courses would help. Yeah,

437
01:41:35.430 --> 01:41:49.979
Chair- Rajeev Alur: you know, co-authored by the you know the organizers and the session chairs, and this would summarize the findings of the workshop, and that would be made publicly available. And this hopefully

438
01:42:05.100 --> 01:42:07.939
Chair- Rajeev Alur: that, uh, we collectively

439
01:42:41.700 --> 01:42:44.590
Chair- Rajeev Alur: follow up with an email on the next steps, and then

440
01:42:44.600 --> 01:42:47.129
Chair- Rajeev Alur: go this to, you know. Come up with some

441
01:42:53.100 --> 01:43:00.980
Chair- Rajeev Alur: you to all the Nsf. Post here, particularly in

442
01:43:10.930 --> 01:43:13.910
Nina Amla: Rajiv that there's a question.

443
01:43:16.640 --> 01:43:33.269
Chair- Rajeev Alur: Yeah. I was about to write an answer to that question. Yeah. So the question is with the most recent failures of autonomous systems. For example, Tesla over accidents.

444
01:43:33.280 --> 01:43:41.110
Chair- Sayan Mitra: I have. We already arrived at the evive bug moment for autonomous systems, And so

445
01:43:41.770 --> 01:43:52.070
Chair- Sayan Mitra: okay, do formal methods have a way to say something interesting about these types of problems in autonomous systems already or not.

446
01:43:54.820 --> 01:43:57.380
Chair- Sayan Mitra: So my take is that

447
01:43:57.440 --> 01:44:13.120

Chair- Sayan Mitra: we have some ideas about why these bugs occurred, for example, in this Uber accident. The problem is widely pinned on perception. The problem was with the computer region system, but

448
01:44:13.850 --> 01:44:27.519
Chair- Sayan Mitra: ultimately we have to reason about the whole system, not just about the individual components, and, as you heard from the Ai part of the discussion, specifying correctness of perception systems is very difficult,

449
01:44:27.580 --> 01:44:35.650
Chair- Sayan Mitra: but that doesn't mean we can just not address this problem, right? So

450
01:44:35.800 --> 01:44:55.380
Chair- Sayan Mitra: I think these failures indicate that formal methods will definitely have an important role to play, particularly when the regulatory environment changes a little bit. But at the same time it's a little early to say that currently we have the tools to address

451
01:44:55.700 --> 01:44:57.849
Chair- Sayan Mitra: all of these things already.

452
01:44:58.250 --> 01:45:00.199
Chair- Sayan Mitra: Others might want to add more to that.

453
01:45:02.100 --> 01:45:03.889
Chair- Rajeev Alur: He actually that big,

454
01:45:30.920 --> 01:45:33.089
Chair- Rajeev Alur: so that it's a kind of firm.

455
01:45:59.790 --> 01:46:16.880
Chair- Ratul Mahajan: If if you're having a discussion on it, and quickly add, like what what really kind of happened in networking, I think, when large accidents happen, like, I think you look for what tools you have, and oftentimes like. Answer is not necessarily a form of methods to like for a lot of networking world

456

01:46:16.890 --> 01:46:46.879
Chair- Ratul Mahajan: the tool would be, Let's automate the network more. Let's take someone out of the just simple automation of work. Then you say, Okay, if it happens again, once you've done that it happens again. We're like, okay. Maybe I should be, you know, pulling some more discipline around how I configure my network. And so, once you've done that again. Then what happened? So I think like you, you so that once you run out of super bullets like that, or what people think it's, you know. Put it then. You kind of you know the form method. Let's try that now. So different industries are in different place.

457
01:46:46.890 --> 01:46:53.439
Chair- Ratul Mahajan: So when an accident happens, there's a belief in like, you know, whatever's easy, achievable as an option. You pick that first,

458
01:46:53.520 --> 01:47:08.029
Chair- Ratul Mahajan: and it does suddenly adds value, whether it solves a problem or not. So I think, like networking, at least of hyper skaters, I think. Ah, we ran out of those types of things at that point you have to get. Then you have to kind of bite the bullet, and you start using formal methods.

459
01:47:08.840 --> 01:47:14.050
Chair- Ratul Mahajan: Just that simple design, discipline and other automation tricks. Basically don't work anymore.

460
01:47:28.270 --> 01:47:29.189
Chair- Rajeev Alur: A good thanks.

461
01:47:29.200 --> 01:47:31.509
Chair- Rajeev Alur: There any any more questions?

462
01:47:33.370 --> 01:47:35.429
Chair- Rajeev Alur: No, he has not.

463
01:47:37.000 --> 01:47:54.939
Chair- Rajeev Alur: Yeah. So there are no more questions. Um, yeah, you know, we would like to thank Reggie first, for, you know, for carrying this whole Ti meeting very nicely, and, you know, putting together

464
01:47:54.950 --> 01:48:11.620
NSF- Pavithra Prabhakar: a great team of session chairs, who handled this very beautifully, and the report outs were great. Um, you know, very informative of information, and we look forward to more details. But but you know, great job, everybody

465
01:48:11.630 --> 01:48:16.960
NSF- Pavithra Prabhakar: running the meetings with all the attendees, and finally, you know,

466
01:48:16.970 --> 01:48:43.579
NSF- Pavithra Prabhakar: for everybody, for attending and making this uh really um interactive discussions during sessions and so on. So we are grateful to all the ti's for uh for making it to this meeting. Um, you know it, despite some people couldn't make it. We had a very good presence and great discussion. So um, yeah, we We look forward to more interactions and the complete report in the coming days.

467
01:48:43.590 --> 01:48:46.739
NSF- Pavithra Prabhakar: And yeah, thanks, everybody again.

468
01:48:46.810 --> 01:48:50.050
NSF- Pavithra Prabhakar: And uh, we are also thankful to our um

469
01:48:50.220 --> 01:48:53.419
NSF- Pavithra Prabhakar: to all the other uh program directors

470
01:48:53.430 --> 01:49:22.119
NSF- Pavithra Prabhakar: for ah giving their ah time. And Ah, you know this! As I said, this is a very multi-dimensional output. Ah, there is the four of us. Ah! From Ccf! But then we have ah several of our program directors coming from Cms and Iis, who have really taken um, you know a very great interest in this program, and have been responsible for ah to feel part of these formal methods initiatives.

471
01:49:22.200 --> 01:49:26.789
NSF- Pavithra Prabhakar: So, thanks to everybody And, Nina, if you want to add any few words,

472
01:49:26.800 --> 01:49:46.670

Nina Amla: I i'm just end with something for all of you to think about right. So when Ranji all I ran this workshop, I think, which got us thinking about this. I don't know a long time ago, I think two thousand and twelve. I don't know if Francis is here on attending, but it. There was a line in there that said formal methods is like broccoli

473
01:49:46.680 --> 01:49:57.369
Nina Amla: clearly very good for you, but rather not to Everyone's liking right? So I'd like you to think a little bit about. How can we change that mindset?

474
01:49:57.400 --> 01:50:13.469
Nina Amla: Right? We don't want formal methods to be everybody's broccoli. We need formal. We need to really think about how we can make this an indispensable part of everyone's toolbox across all fields. So i'll leave you with that, because you guys are the ones who are going to make that happen.

475
01:50:13.480 --> 01:50:31.610
Nina Amla: But I hope never to see Broccoli showing up in any of our workshop reports going forward with respect to home methods, and we also Thank everybody. Thank you so much for participating. Thank you so much for organizing. It was really really a really fun,

476
01:50:32.380 --> 01:50:33.960
Chair- Elena Glassman: this is

477
01:50:34.520 --> 01:50:35.550
It's a

478
01:50:37.370 --> 01:50:40.249
Chair- Rajeev Alur: thanks, everyone. Thank you very much.