

# Logic and Computation – CS 2800

## Fall 2019

### Lecture 36

### Parting thoughts

Stavros Tripakis



**Northeastern University**  
**Khoury College of**  
**Computer Sciences**

# Outline

- Where we have arrived
- Where to go next

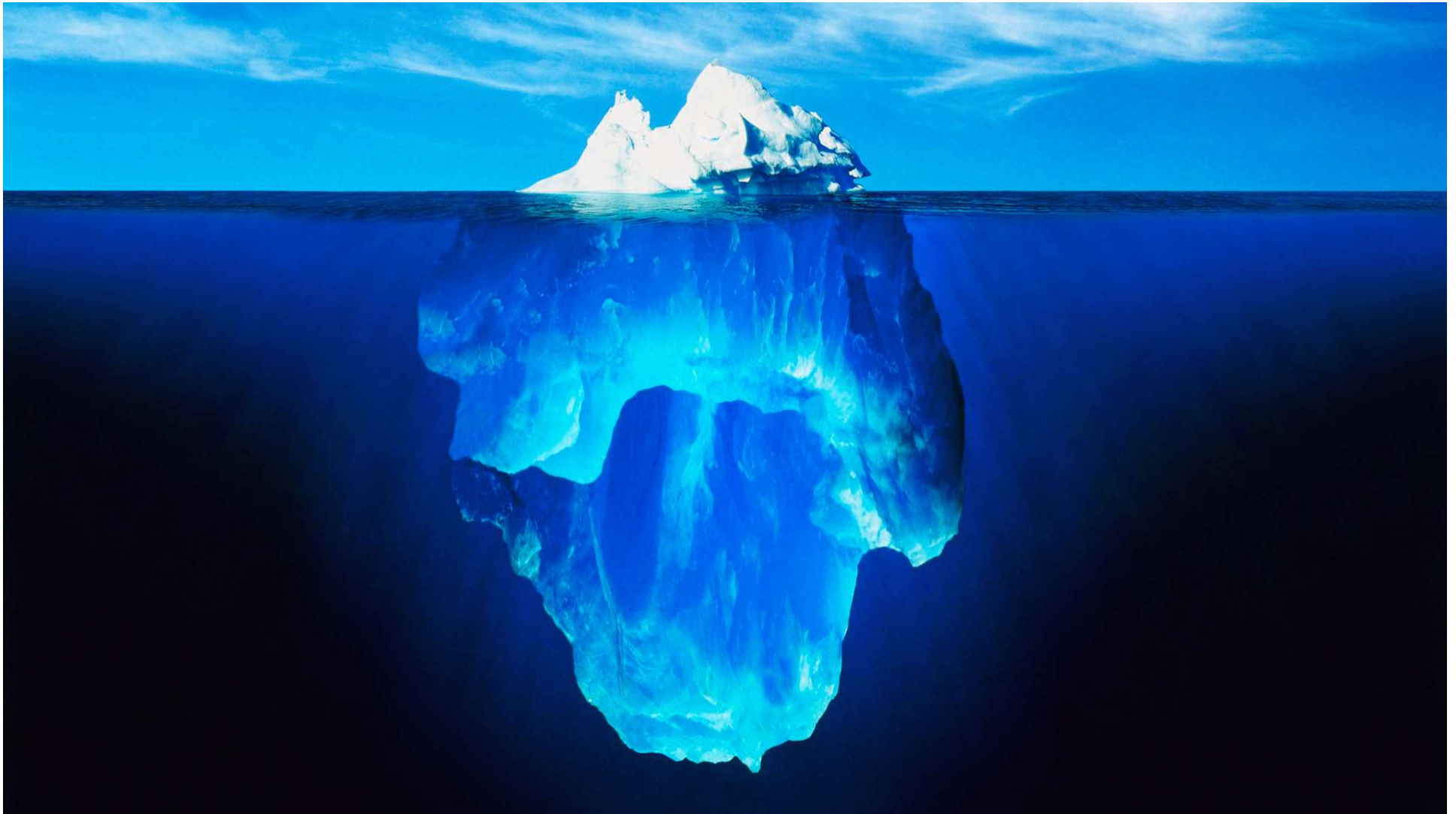
It's been an interesting journey

# You should feel good about being able to

- Write programs that express precisely their input/output requirements = write **contracts**!
- Write down precisely what you want your program to do = write down **formal properties** about programs!
- Use a tool that can automatically generate tests to check, and sometimes also prove, these properties.
- **Prove** the properties yourself = make them **theorems**!
  - Equational proofs, induction, termination, measure functions, invariants, inductive invariants, ...

# Where to go next

# The tip of the iceberg



# Courses that you can take

- Formal methods, verification:
  - CS 4820 *Computer-Aided Reasoning*
  - CS 4830 *System Specification, Verification and Synthesis:*  
<http://www.ccs.neu.edu/~stavros/ssvs19.html>
  - CS 7485 *Special Topics in Formal Methods*
  - ...
- Programming languages, type theory, ...

# CS 4830 *System Specification, Verification and Synthesis*

- How to **model** software and other **systems** (e.g., digital circuits, multi-threaded programs, network protocols, voting machines, banking systems, databases, distributed systems, satellites, Mars rovers, self-driving cars, robots, pacemakers, ...)?
- How to **specify** properties about such systems?
- How to **verify** (prove) those properties?
- Can we **synthesize** correct systems **automatically**?!
- ...



## CONTRIBUTED ARTICLES



## How Amazon Web Services Uses Formal Methods

By Chris Newcombe, Tim Rath, Fan Zhang, Bogdan Munteanu, Marc Brooker, Michael Deardeuff

Communications of the ACM, Vol. 58 No. 4, Pages 66-73

10.1145/2699417

[Comments \(1\)](#)

VIEW AS:       SHARE:    



Since 2011, engineers at Amazon have used formal specification and model checking to solve design problems in critical systems. This paper discusses the motivation and experience, the domain, and what has not been done. We refer to the authors by their first names.

At AWS we strive to build services that are simple to use. External simplicity is achieved by abstracting complex distributed systems.

### Key Insights

- **Formal methods find bugs in system designs that cannot be found through any other technique we know of.**
- **Formal methods are surprisingly feasible for mainstream software development and give good return on investment.**
- **At Amazon, formal methods are routinely applied to the design of complex real-world software, including public cloud services.**

Απ. Καλδάρας, Ένα τραγούδι X Inbox (142) - stavros.tripakis@... X Google Calendar - December 2 X Paxos Algorithm X

https://lamport.azurewebsites.net/tla/paxos-algorithm.html?back-link=news.html

myNEU Poll Ev login pollev Stavros Tripakis Khoury Admin Jazz 24/7 Radio Blackboard CS 2800 CS 2800 CS 4830/7485 InvGenGame Jupiter JTS789-787 Te...

# The Paxos Algorithm or How to Win a Turing Award

Leslie Lamport  
*Last modified on 23 August 2019*

[Back](#)

[Home](#)

[High-Level View](#)

[News](#)

[Industrial Use](#)

[Learning](#)

[The Toolbox](#)

[Tools](#)

[Advanced Topics](#)

In July 2019 I gave a pair of 1-1/2 hour lectures at a summer school on distributed computing in Saint Petersburg, Russia. The lectures covered three topics:

- The Paxos consensus algorithm.
- TLA+
- How to win a Turing award.

If you are interested in any of those topics, you might want to watch the videos of the lectures. In them, I present a rigorous development of the Paxos consensus algorithm that formalizes the thinking I believe led me to discover it. I explain the algorithm with three specifications: a trivial one stating what the algorithm is supposed to accomplish, a high-level algorithm in which each process directly views the state of every other process, and the actual algorithm in which processes communicate by sending messages.

The lectures assume no prior knowledge of the Paxos consensus algorithm or of TLA+, the language in which the specifications are written. In fact, the lectures constitute a crash course on TLA+. If you're interested just in the algorithm, you will probably find the specs annoying and will try to read only the plentiful comments, ignoring the TLA+. To understand the algorithm, you will have to stop the video at certain points and read


Type here to search

9:03 AM  
12/4/2019

Απ. Καλδάρας, Ένα τραγ... | Inbox (142) - stavros.tripakis... | Google Calendar - December... | Paxos Algorithm | Spin - Formal Verification

spinroot.com/spin/whatispin.html

myNEU | Poll Ev | login pollev | Stavros Tripakis | Khoury Admin | Jazz 24/7 Radio | Blackboard | CS 2800 | CS 2800 | CS 4830/7485 | InvGenGame | Jupiter JTS789-787 Te...



**Verifying Multi-threaded Software with Spin**

**Spin** is a popular open-source software verification tool, used by thousands of people worldwide. The tool can be used for the formal verification of multi-threaded software applications. The tool was developed at [Bell Labs](#) in the Unix group of the Computing Sciences Research Center, starting in 1980. The software has been available freely since 1991, and continues to evolve to keep pace with new developments. In April 2002 the tool was awarded the ACM [System Software Award](#). [\[read more\]](#)

<b>discover</b>	<b>learn</b>	<b>use</b>	<b>community</b>
<ul style="list-style-type: none"> <li>• <a href="#">what is spin?</a></li> <li>• <a href="#">success stories</a></li> <li>• <a href="#">examples</a></li> <li>• <a href="#">roots</a></li> </ul>	<ul style="list-style-type: none"> <li>• <a href="#">tutorials</a></li> <li>• <a href="#">books</a></li> <li>• <a href="#">papers</a></li> <li>• <a href="#">model extraction</a></li> <li>• <a href="#">exercises</a></li> </ul>	<ul style="list-style-type: none"> <li>• <a href="#">installation</a></li> <li>• <a href="#">man pages</a></li> <li>• <a href="#">options</a></li> <li>• <a href="#">releases</a></li> </ul>	<ul style="list-style-type: none"> <li>• <a href="#">forum</a></li> <li>• <a href="#">symposia</a></li> <li>• <a href="#">support</a></li> <li>• <a href="#">projects</a></li> </ul>

**Open Source:** Starting with Version 6.4.5 from January 2016, the Spin sources are available under the standard BSD 3-Clause open source license. Spin is now also part of the latest stable release of Debian

Type here to search | 9:04 AM 12/4/2019

# TRACE evaluations

- **Surveys are anonymous**
- Please respond to the survey!
- Surveys close on Dec 13