Logic in computer science, engineering, industry and (time permitting) in math

Yuri Gurevich, Prof. Emeritus
Computer Science & Engineering
University of Michigan

Logic

Began as a tool of convincing argumentation, along with rhetoric, demagoguery, theatrics. Branched out into several areas, in particular:

- Statistical reasoning
- Judicial logic, with various legal standards for the burden of proof: by preponderance of evidence, clear and convincing evidence, beyond reasonable doubt.
- Math logic uses math methods and arguably studies the logic of math. Armchair arguing is easier than that in the court of law.

Mathematical logic

- Prehistory
  - Aristotle, Boole, Frege
  - Russell's paradox: $x \in R \Longleftrightarrow x \notin x$
- Foundation of mathematics
- Formal languages

LOGIC IN CS

Types

- History
  - Russell and Whitehead
- Nowadays
  - Programming languages
  - Java virtual machine, .Net
  - Static analysis of programs

Recursion

- History
  - Hilbert, Ackerman and Rózsa Péter
  - Gödel’s recursive calculus for general algorithms
- Nowadays
  - Recursion theory
  - Functional programming languages
  - Syntax and semantics of formal languages

- Much of what logicians thought about earlier on has found applications in CS.

Machine models

- Turing machines
  - Formalization of the notion of algorithm
  - Universal algorithms
- Von Neumann architecture
- Random access machines
- Cellular automata, neural networks

Complexity theory

- One precursor: Constructive math
- Time and space complexity classes
- P =? NP
  - Steve Cook: analogy with recursive vs. recursively enumerable
  - Leonid Levin, and the problem of perebor, i.e. exhaustive-search.

Model theory to database theory

- First-order logic
- Relational databases
  - First-order structures. Schemas, with their attributes, are improved vocabularies.
  - Codd’s operations vs. Tarski’s cylindrical algebra
- Implementation independence, and polynomial time for general structures, e.g. unordered graphs.

Proof theory then and now

- Classical vs. intuitionistic deductive systems
- Theorem provers, e.g. Coq
- SAT solvers, SMT (satisfiability modulo theories) solvers

One under-appreciated story

Solve the following "proportion":

$$\frac{FSA}{Reg} \propto \frac{NPDA}{CF} \propto \frac{DPDA}{X}$$

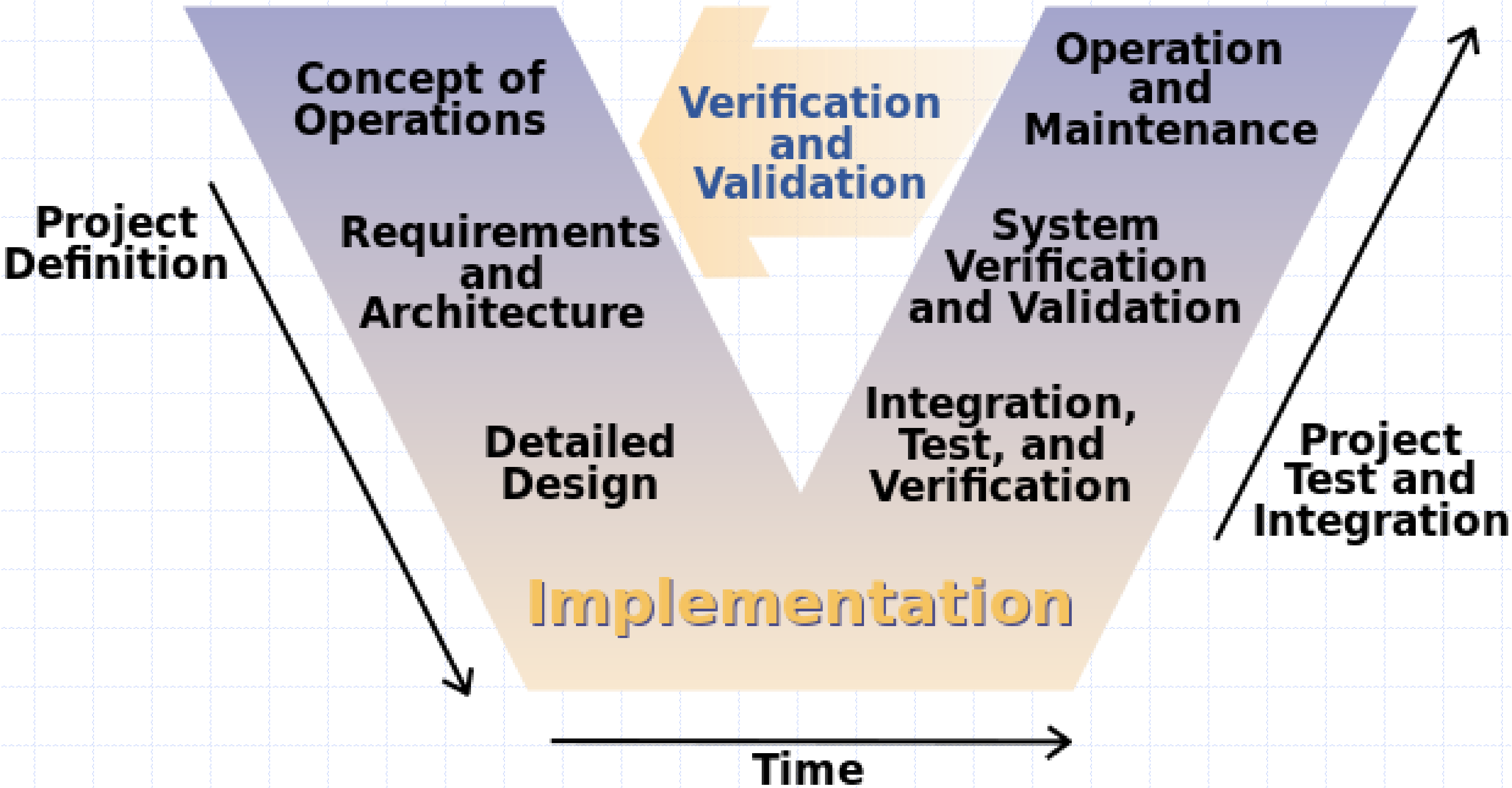
Donald Knuth did. LR(k) languages.

This is about formal languages but with little forerunner logic work.

Software specifications

- The issue of software specification is another example of the use of logic in computer science that has little forerunner logic work.
- This is the issue that brought me from the University of Michigan to Microsoft.
- While there has been much work in the area, I take the liberty to go more personal on this one particular issue.

The V-diagram



From Wikipedia

Personal story

- 1982, UM
  - What's Pascal? Semantics of programs
  - What is computer science about?
- What's an algorithm
  - Declarative/imperative
  - High-level and executable; is that possible?
  - Abstract state machines
- 1998, Microsoft
  - Specifications and model-based testing
  - Architects, developers, testers, and researchers
- Spec Explorer; EU vs. Microsoft

LOGIC IN CE

- History
  - Boolean circuits
  - Many-valued logics
- What drives the use of logic nowadays? Formal languages.

Proliferation of formal languages

- Programming languages
  - Machine languages
  - Assembly languages
  - C, C++, C#, Java, …
- Database languages
- Specification languages
- Authentication/authorization languages
- Hardware languages, like Verilog
- Languages for quantum computing

Engineers do logic day in day out

- Programming in different languages at the bewildering variety of levels of abstraction
- Writing compilers. A compiler for a language L is a universal program for L (possibly though not necessarily in L).
- Writing specifications, verification of specifications
- Testing
  - Model based testing
  - Conformance testing

Logic day in day out (cont.)

- Formalizing stuff e.g. certificates, claims
- Creating specialized languages, e.g. XACML, eXtensible Access Control Markup Language
- Model checking, i.e. automatic verification of finite-state-machines properties
- Increasing use of assertion verification, SAT solvers, provers

Information leakage problem

- The info leakage problem is a good example of a confusing medley of abstraction levels.
- There are numerous off-the-shelf tools working at different levels of abstraction. This helps but does not solve the problem.

Software engineers do not know logic

- Very few studied logic.
  Instead they studied calculus which they rarely, if ever, use.
- Even the brightest of them – who may be brilliant – don't know formal logic.
- Typically they do not realize even that there is a science of logic that is relevant to their work.
- From a conversation with a talented software architect: "I guess their language is a subset of yours."

The syntax divide

- Engineers’ thinking is typically very syntactical.
  - Code is the meaning.
- Logicians always speak about formulas but don’t write them honestly.
- Precise vs “precisable.” Logicians often are cavalier about things that are clear in principle.

Feasibility vs complexity

- We, the logicians, long neglected practical complications of propositional logic.
  - "Without loss of generality φ is in conjunctive normal form."

The semantic divide

- This is where logicians shine, and engineers are cavalier to their own peril.
- The price that they – and society! – pay is big.
- However logicians live in a much cleaner world.
  - It is not enough that software has the right functionality. It should have good performance, be maintainable, be legacy compatible, etc.
  - The correctness of software

Declarative/imperative divide

- In the theory of abstract state machines, structures (normally static in logic) evolve as computations progress
- In authorization, there are communication rules and filters. E.g.

if α then send [with justification] β to p,

if α then accept [with justification only] ***pattern*** **from p.**

- Obligations have imperative aspects.

So what is needed most?

- Logical literacy
- Understanding both hazards of abstraction: under-abstraction and over-abstraction
- Semantics and the interplay of syntax and semantics

Remark. It is soundness that is most needed by engineers. Completeness is typically too good to be true.

LOGIC IN MATH

Diverse history

- USA. Logicians have been fighting to achieve the acceptance of mathematicians.
- Europe on the ETHZ Math Dept example. Zermelo, Weyl, Gonseth, Bernays, Specker, Lauchli, Engeler. Now: nobody.
- Russia. Formal logic was virtually forbidden in the 1930s. After the 1960s thaw, it was held in high regard. Kolmogorov chaired the MSU logic department until his death.

What does the future hold?

- There is much inertia in the academy. In the long run much depends on whether logicians contribute to math at large.
- To contribute, the logician should know the relevant math intimately.
- But what is easier, for a logician to learn relevant math, or for a mathematician to learn relevant logic?

Example: Lefschetz's principle

- Tarski, but – as far as the math is concerned - also Chevalley (constructible sets are closed under projections).
- Barwise - Eklof
- "Although I am aware of the precise formulations using first order logic and beyond ..., I tend not to use them. Rather I view the Lefschetz principle as more of a philosophical principle of what ought to be possible in general, and do the necessary verifications as and when I need them ... I suspect this attitude is pretty common among many algebraic geometers," Donu Arapura.

Example: Kurt Gödel and Pierre Deligne

Another example of mathematicians rediscovering a logic theorem:

- Every coherent topos has enough points.

Example: Whitehead problem

- Is every Whitehead abelian group A free?
  - Whitehead: If $f: B \to A$ is surjective with kernel **Z**, then there is $g: A \to B$ with $fg = 1_A$.
- Shelah
  - V=L → positive
  - Martin's axiom + ¬CH → negative

A mathematician can work with such tools without going deeply into forcing.

Calkin Algebra Problem

- Are there outer automorphisms?
  - Calkin Algebra: The quotient of the ring of bounded linear operators on a separable infinite-dimensional Hilbert space by the ideal of compact operators.
- CH → Yes (Phillips & Weaver, 2007)
- Proper Forcing Axiom → No (by Farah, 2011)

Understanding Shelah's proper forcing is a nontrivial time investment for a mathematician.

THANK YOU