Proofs and Things

N. Shankar

Computer Science Laboratory
SRI International
Menlo Park, CA

May 23, 2011

Welcome to Summer Formal 2011

- The school is an outgrowth of a workshop during Fall 2010 on Usable Verification initiated by *Lenore Zuck*, and organized with *Tom Ball*.
- We are grateful to her, *Helen Gill*, *Sol Greenspan*, *Nina Amla*, and *Sam Weber* at the US National Science Foundation for their enthusiastic sponsorship.
- At SRI, *Pat Lincoln*, *John Rushby*, and *Sam Owre* have been pillars of encouragement and support.
- We also thank *Joe O'Brien* (Events) and *Eric Drake* (Catering) at Menlo College.

Outline

- We take a compressed look at the past, present, and future of formal techniques.
- Formality has been around in one form or another since ancient times.
- The mid-to-late nineteenth century saw the world of mathematics come to grips with the formal foundations of mathematics.
- The first half of the twentieth century yielded deep advances in these formal foundations.
- With the advent of powerful computing machines, many powerful formal reasoning techniques were developed in the second half of the twentieth century.
- These techniques are not only relevant for building reliable software, they also have useful applications in other disciplines.

Overview

- Proofs have a curious history spanning over two millennia.
- First, people observed certain patterns in nature.
- These observations were crystallized into abstractions such as counting, grouping, ordering, partitioning, transforming.
- Slowly, they realized that these patterns could be *explained*, and . . .
- The rules underlying such explanations could be codified and made rigorous by means of proof.
- Proofs became indispensable for exposing fallacies and for reasoning beyond intuition.
- *Formal* proofs captured the basic rules of the game thus allowing *calculation* and *metamathematics*.
- In the twenty-first century, proof technology is being applied to complex virtual and physical systems in the real world.
-

A Small Puzzle [Wason]

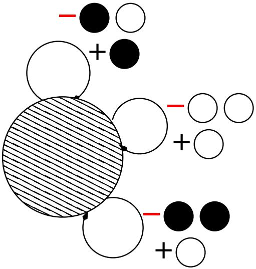
- Given four cards laid out on a table as: $\boxed{\text{D}}$, $\boxed{3}$, $\boxed{\text{F}}$, $\boxed{7}$, where each card has a letter on one side and a number on the other.
- Which cards should you flip over to determine if every card with a $\boxed{\text{D}}$ on one side has a $\boxed{7}$ on the other side?

A Small Problem

Given a bag containing some black balls and white balls, and a stash of black/white balls. Repeatedly

1. Remove a random pair of balls from the bag
2. If they are the same color, insert a white ball into the bag
3. If they are of different colors, insert a black ball into the bag

What is the color of the last ball?



The Monty Hall Problem

- There are three doors with a car behind one, and goats behind the other two.
- You have chosen one door.
- Monty Hall, knowing where the car is hidden, opens one of the other two doors to reveal a goat.
- He allows you to switch your choice to the other closed door.
- If you want to win the car, should you switch?

Gilbreath's Card Trick

- Start with a deck consisting of a stack of quartets, where the cards in each quartet appear in suit order $\spadesuit, \heartsuit, \clubsuit, \diamondsuit$:

$$\begin{array}{c}\langle 5\spadesuit\rangle, \langle 3\heartsuit\rangle, \langle Q\clubsuit\rangle, \langle 8\diamondsuit\rangle,\\ \langle K\spadesuit\rangle, \langle 2\heartsuit\rangle, \langle 7\clubsuit\rangle, \langle 4\diamondsuit\rangle,\\ \langle 8\spadesuit\rangle, \langle J\heartsuit\rangle, \langle 9\clubsuit\rangle, \langle A\diamondsuit\rangle\end{array}$$

- Cut the deck, say as $\langle 5\spadesuit\rangle, \langle 3\heartsuit\rangle, \langle Q\clubsuit\rangle, \langle 8\diamondsuit\rangle, \langle K\spadesuit\rangle$ and $\langle 2\heartsuit\rangle, \langle 7\clubsuit\rangle, \langle 4\diamondsuit\rangle, \langle 8\spadesuit\rangle, \langle J\heartsuit\rangle, \langle 9\clubsuit\rangle, \langle A\diamondsuit\rangle$.
- Reverse one of the decks as $\langle K\spadesuit\rangle, \langle 8\diamondsuit\rangle, \langle Q\clubsuit\rangle, \langle 3\heartsuit\rangle, \langle 5\spadesuit\rangle$.
- Now shuffling, for example, as

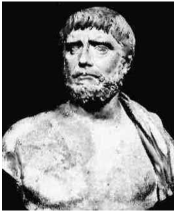
$$\begin{array}{c}\langle 2\heartsuit\rangle, \langle 7\clubsuit\rangle, \underline{\langle K\spadesuit\rangle}, \underline{\langle 8\diamondsuit\rangle},\\ \langle 4\diamondsuit\rangle, \langle 8\spadesuit\rangle, \underline{\langle Q\clubsuit\rangle}, \langle J\heartsuit\rangle,\\ \underline{\langle 3\heartsuit\rangle}, \langle 9\clubsuit\rangle, \underline{\langle 5\spadesuit\rangle}, \langle A\diamondsuit\rangle\end{array}$$

- *Each quartet contains a card from each suit.*

Proving Gilbreath [Hoare–S]

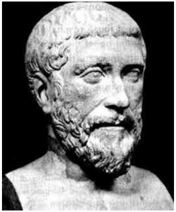
- *Why does the trick always work?*
- Each quartet from the shuffled deck contains k cards from the first deck and $4 - k$ cards from the top of the reversed deck.
- So we can skip the shuffle and take k cards from the top of the first deck and the bottom of the unreversed deck.
- The, we can also skip the cut and take k cards from the top and $4 - k$ cards from the bottom of the original uncut deck.
- The remainder of the deck with $n - 4$ cards is quartet-wise ordered by suit, and the pattern repeats (*induction*).

In the Beginning



Thales of Miletus (624 – 547 B.C.): Earliest known person to be credited with theorems and proofs.

> *It was Thales who first conceived the principle of explaining the multitude of phenomena by a small number of hypotheses for all the various manifestations of matter.*



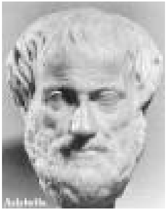
Pythagoras of Samos (569–475 B.C.): Systematic study of mathematics for its own sake.

> *... he tried to use his symbolic method of teaching which was similar in all respects to the lessons he had learnt in Egypt. The Samians were not very keen on this method and treated him in a rude and improper manner.*

The Axiomatic Method

Plato (427–347 B.C.): Suggested the idea of a single axiom system for all knowledge.

> *the reality which scientific thought is seeking must be expressible in mathematical terms, mathematics being the most precise and definite kind of thinking of which we are capable.*



Aristotle (384–322 B.C.) of Stagira: Laid the foundation for scientific thought by proposing that all theoretical disciplines must be based on axiomatic principles.

The Elements

Euclid of Alexandria (325–265 B.C.): Systematic compilation and exposition of geometry and number theory.

1. A straight line segment can be drawn joining any two points.
2. Any straight line segment can be extended indefinitely in a straight line.
3. Given any straight line segment, a circle can be drawn having the segment as radius and one endpoint as center.
4. All right angles are congruent.
5. If two lines are drawn which intersect a third in such a way that the sum of the inner angles on one side is less than two right angles, then the two lines inevitably must intersect each other on that side if extended far enough. This postulate is equivalent to what is known as the parallel postulate.

A Glimmer of Rationality

Ramon Llull (1235–1316): Talked of *reducing all knowledge to first principles*. Developed a symbolic notation (Ars Magna) and conceived of a reasoning machine.

> *When he attempted to apply rational thinking to religion, Pope Gregor XI "accused him of confusing faith with reason and condemned his teachings."*

Gottfried Leibniz (1646–1716) The idea of a formal language (*characteristica universalis*) for expressing scholarly knowledge and a mechanical method for making deductions (*calculus ratiocinator*).

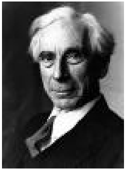
> *What must be achieved is in fact this: that every paralogism be recognized as an* error of calculation, *and every* sophism *when expressed in this new kind of notation, appear as a* solecism *or* barbarism, *to be corrected easily by the laws of this philosophical grammar.*
>
> *Once this is done, then when a controversy arises, disputation will no more be needed between two philosophers than between two computers. It will suffice that, pen in hand, they sit down to their abacus and (calling in a friend, if they so wish) say to each other:* let us calculate.

The Age of Reason

George Boole (1815–1864): Algebraic system for propositional reasoning. A major turning point: admitted systematic calculation into logic.

Modern Logic

Gottlob Frege (1848–1925): A system of quantificational logic.



Bertrand Russell (1872–1970) and Alfred North Whitehead (1861–1947): Rigorous formal development of a significant portion of mathematics.

Modern Logic

David Hilbert (1862–1943): Consistency of mathematics could perhaps be verified by *meta-mathematical* methods applied to formal logic.

> *In mathematics there is no ignorabimus.*
>
> *We must know — we will know!*

Kurt Gödel (1906–1978): Completeness: Every statement has a counter-model or a proof.
Incompleteness: Any consistent formal theory for arithmetic contains statements that are neither provable nor disprovable.
Such a theory cannot prove its own consistency.

Computability

Jacques Herbrand (1908–1931): Suggested a (valid) definition of computability to Gödel; Herbrand's theorem is the central result in automated theorem proving.

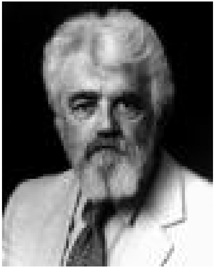
Alonzo Church (1909–1995): Introduced lambda calculus, ls Church's thesis, and examples of unsolvable problems.

Alan Turing (1912–1954): Formal definition of calculability: the Turing machine, universal Turing machine, unsolvability of halting problem.

Vannevar Bush [1890–1974]: As We May Think

Logic can become enormously difficult, and it would undoubtedly be well to produce more assurance in its use. . . . We may some day click off arguments on a machine with the same assurance that we now enter sales on a cash register.

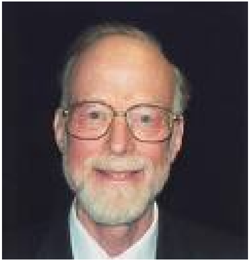
John McCarthy: Logic and Artificial Intelligence



Checking mathematical proofs is potentially one of the most interesting and useful applications of automatic computers.

Tony Hoare and Edsger Dijkstra [1930–2002]

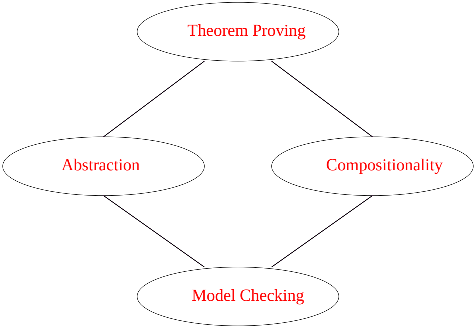
Logical Foundations of Program Correctness



By evoking the need for deep conceptual hierarchies, the automatic computer confronts us with a radically new intellectual challenge that has no precedent in our history . . . it is no longer the purpose of programs to instruct our machines; these days, it is the purpose of machines to execute our programs.

Amir Pnueli [1941–2009]

Temporal Logic and Algorithmic Verification



The Sequent Calculus

	Left	Right
Ax	$\frac{}{\Gamma, A \vdash A, \Delta}$	
$\neg$	$\frac{\Gamma \vdash A, \Delta}{\Gamma, \neg A \vdash \Delta}$	$\frac{\Gamma, A \vdash \Delta}{\Gamma \vdash \neg A, \Delta}$
$\vee$	$\frac{\Gamma, A \vdash \Delta \quad \Gamma, B \vdash \Delta}{\Gamma, A \vee B \vdash \Delta}$	$\frac{\Gamma \vdash A, B, \Delta}{\Gamma \vdash A \vee B, \Delta}$
$\wedge$	$\frac{\Gamma, A, B \vdash \Delta}{\Gamma, A \wedge B \vdash \Delta}$	$\frac{\Gamma \vdash A, \Delta \quad \Gamma \vdash B, \Delta}{\Gamma \vdash A \wedge B, \Delta}$
$\Rightarrow$	$\frac{\Gamma, B \vdash \Delta \quad \Gamma \vdash A, \Delta}{\Gamma, A \Rightarrow B \vdash \Delta}$	$\frac{\Gamma, A \vdash B, \Delta}{\Gamma \vdash A \Rightarrow B, \Delta}$
$\forall$	$\frac{\Gamma, A[t/x] \vdash \Delta}{\Gamma, \forall x : A \vdash \Delta}$	$\frac{\Gamma \vdash A[c/x], \Delta}{\Gamma \vdash \forall x : A, \Delta}$
$\exists$	$\frac{\Gamma, A[c/x] \vdash \Delta}{\Gamma, \exists x : A \vdash \Delta}$	$\frac{\Gamma \vdash A[t/x], \Delta}{\Gamma \vdash \exists x : A, \Delta}$
Cut	$\frac{\Gamma \vdash A, \Delta \quad \Gamma, A \vdash \Delta}{\Gamma \vdash \Delta}$	

Peirce's Formula

- A sequent calculus proof of Peirce's formula $((p \Rightarrow q) \Rightarrow p) \Rightarrow p$ is given by

$$\dfrac{\dfrac{\dfrac{\dfrac{}{p \vdash p, q}\,Ax}{\vdash p, p \Rightarrow q}\,\vdash\Rightarrow \qquad \dfrac{}{p \vdash p}\,Ax}{(p \Rightarrow q) \Rightarrow p \vdash p}\,\Rightarrow\vdash}{\vdash ((p \Rightarrow q) \Rightarrow p) \Rightarrow p}\,\vdash\Rightarrow$$

Reasoning Methods: Saturation/Resolution

We want to show that $\neg p \vee \neg q \vee r,\quad \neg p \vee q,\quad p \vee r,\quad \neg r$ is unsatisfiable.

$$\begin{array}{c}
(K_0 =)\ \neg p \vee \neg q \vee r,\quad \neg p \vee q,\quad p \vee r,\quad \neg r \\ \hline
(K_1 =)\ \neg q \vee r,\quad K_0 \\ \hline
(K_2 =)\ q \vee r,\quad K_1 \\ \hline
(K_3 =)\ r,\quad K_2 \\ \hline
\bot
\end{array}
\begin{array}{l}
\\ \textbf{Res} \\ \textbf{Res} \\ \textbf{Res} \\ \textbf{Contrad}
\end{array}$$

Reasoning Methods: DPLL

- The Davis–Putnam–Logemann–Loveland approach works by *propagation* and *search* to complete a partial assignment into a total satisfying assignment.
- For example, the earlier example $\neg p \vee \neg q \vee r, \quad \neg p \vee q, \quad p \vee r, \quad \neg r$ is unsatisfiable by propagation.
- Whereas the variant $\neg p \vee \neg q, \quad \neg p \vee q, \quad p \vee r, \neg r \vee p$ requires search on a truth assignment followed by propagation.
- If we try $r = \top$, then propagation yields $p = \top$, $q = \bot$, and a contradiction (conflict) with $\neg p \vee q$.
- *Analyzing* the conflict yields the *lemma* $\neg p$ and the partial assignment to r can be retracted.
- Now, propagation yields a conflict with r, implying the unsatisfiability of the original constraints.

Reasoning Methods: Theory Solvers

- **Uninterpreted Terms:** $f(f(f(x))) = f(x) \Rightarrow f(f(f(f(f(x))))) = f(x)$.
- **Difference Constraints:** $x - y \leq 1, y - z \leq 1, z - x \leq -3$.
- **Linear Arithmetic:** $x, y, z \geq 0, x + y \leq 2, y - z \geq 3$
- **Arrays:** $i \neq j, A[i := v](j) \neq A(j)$
- **Bit Vectors:** $X \oplus X = 0$

Reasoning Methods: Theory Satisfiability

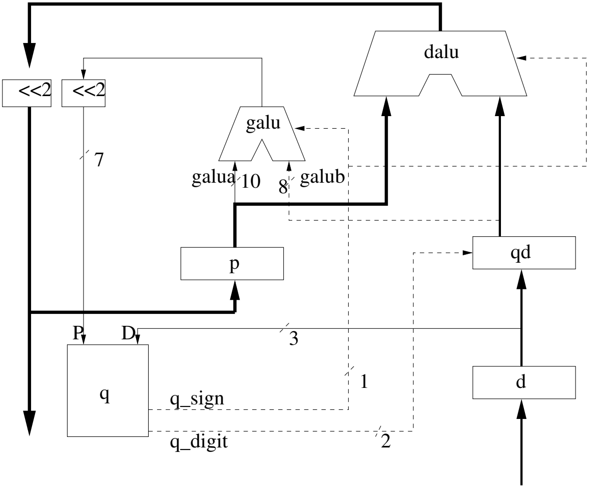
- SMT deals with formulas with theory atoms like $x = y$, $x \neq y$, $x - y \leq 3$, and $A[i := v](j) = w$.
- The DPLL search is augmented with a *theory state* S in addition to the partial assignment.
- Total assignments are *checked* for theory satisfiability.
- When a literal is added to M by propagation, it is also *asserted* to S.
- When a literal is implied by S, it is added to M.
- When backjumping, the literals deleted from M are retracted from S.

Symbolic Model Checking: State Space Explosion

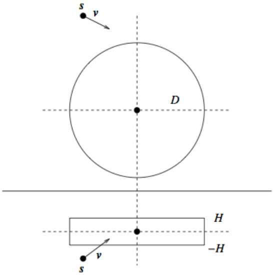
- Mutual exclusion allows resources (e.g., message buffers, queues, devices) to be shared without confusion.
- The processes competing for the resources go from *sleeping* to *waiting* to *critical*, and back to *sleeping*.
- In Peterson's algorithm, each process has a *ready* flag and synchronizes on a *turn* flag.
- Each process enters the critical section unless the other process is ready and holds the flag.
- State space is finite.
- *Is mutual exclusion satisfied?*
- *Can a waiting process be guaranteed entry to the critical section?*

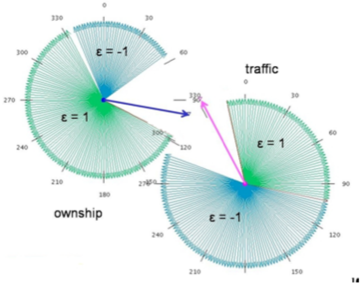
Hardware Verification: Billion Dollar Bugs

The Intel FDIV bug was caused by incorrect entries in the quotient lookup table in an SRT (Sweeney, Robertson, Tocher) divider circuit.



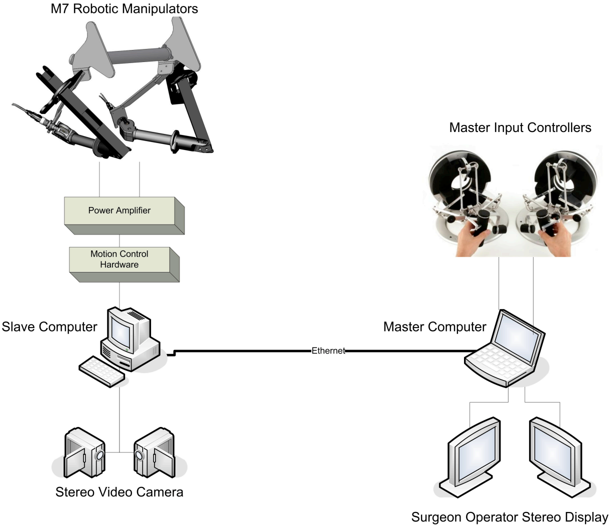
Air-Traffic Control: Collision Detection/Resolution



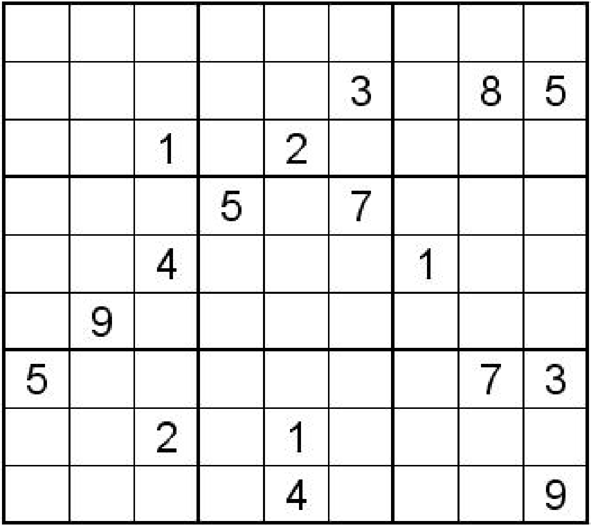


Algorithms and policies comprehensively verified in PVS.

Telesurgery Robots



Hard Sudoku [Wikipedia/Algorithmics_of_Sudoku]



					3		8	5
		1		2				
			5		7			
		4				1		
	9							
5							7	3
		2		1				
				4				9

Hard Sudoku Solved with `sal-inf-bmc` [Whalen]

9	8	7	6	5	4	3	2	1
2	4	6	1	7	3	9	8	5
3	5	1	9	2	8	7	4	6
1	2	8	5	3	7	6	9	4
6	3	4	8	9	2	1	5	7
7	9	5	4	6	1	8	3	2
5	1	9	2	8	6	4	7	3
4	7	2	3	1	9	5	6	8
8	6	3	7	4	5	2	1	9

Proofs of Things

- Proofs may be rigorous and abstract mathematical constructions
- But they can be applied to real things, from card tricks and Sudoku to robots and planes.
- Who needs proof? *We do.*
- What needs proving? *Anything of consequence.*
-

Summer Formal

- Tools: Interactive Theorem Provers (Shankar) and SMT Solvers (de Moura & Dutertre)
- Techniques: Abstraction, Interpolation, Composition (McMillan); Static Analysis (Monniaux)
- Applications: Hardware Verification (Baumgartner); Software model checking and symbolic execution (Rungta and Mehlitz)
- *Hands-on use of different systems — this is a reality show.*

Why a Summer School?

Your Picture Here