

THE IMPORTANCE OF BEING RIGHT

SERGEI ARTEMOV, CUNY GRADUATE CENTER

COMPUTER SCIENCE MIXTER AT CCNY, MAY 8, 2008

COMPUTER BUGS

Computer bugs cost about \$60 billion annually in the US alone. About a third of that cost could be eliminated by improving testing and verification.

SOME FAMOUS COMPUTER BUGS

London ambulance system (1992). A succession of software engineering failures, especially in project management, caused two failures of London's (England) ambulance dispatch system.

The repair cost was estimated at £9m, but it is believed that people died who would not have died if ambulances had reached them as promptly as they would have without the failures.

SOME FAMOUS COMPUTER BUGS

Pentium FDIV bug (1994). Cost Intel half a billion, and a lot of agony on the way to an eventual no-strings-attached recall.

SOME FAMOUS COMPUTER BUGS

Ariane 5 (1996). The Ariane 5 rocket exploded on its maiden flight in June 4, 1996 because the navigation package was inherited from the Ariane 4 without proper testing.






SOME FAMOUS COMPUTER BUGS

USS Yorktown (1998). A crew member of the guided-missile cruiser USS Yorktown mistakenly entered a zero for a data value, which resulted in a division by zero. The error cascaded and eventually shut down the ship's propulsion system. The ship was dead in the water for several hours because a program didn't check for valid input.

SOME FAMOUS COMPUTER BUGS

Mars Climate Orbiter (1999). The 125 million dollar Mars Climate Orbiter was lost by NASA. One of the development teams used Imperial measurement while the other used the metric system of measurement.

SOME FAMOUS COMPUTER BUGS

-  *fighter jets over Dead Sea*
-  *F-16 crossing the Equator,*
-  *Space Shuttle automated landing program,*
-  *another Mars probe - rounding error*
-  *etc.*

CAN COMPUTERS THINK?



CAN COMPUTERS THINK?

Not really!

CAN COMPUTERS THINK?

Not really!

Given a sentence, find its proof:

-  *undecidable for general purpose quantified languages*
-  *unfeasible for general propositional languages*

CAN HUMANS VERIFY?

CAN HUMANS VERIFY?

Not really!

CAN HUMANS VERIFY?

Not really!

- *The aforementioned list of bugs,*
- *an array of notorious erroneous `proofs in Math*
- *years and years to check the correctness of submitted papers in journals, yet with inconclusive results*
- *etc.*

CAN COMPUTERS VERIFY?



CAN COMPUTERS VERIFY?

Yes!

CAN COMPUTERS VERIFY?

Yes!

Given S and p , certify that p is a proof of S :

-  *decidable and feasible for many general purpose languages*
-  *practical, implemented in a variety of computer-based proof assistants*

GENERAL PURPOSE PROOF ASSISTANTS

Prehistory: de Bruijn's Automath Project Modern architecture: Robin Milner (1972) Stanford LCF (Logic for Computable Functions). Circa 1979 - Edinburgh's LCF - tactics, isolated trusted core, proof checker: HOL, Coq, Mizar, Isabelle, PVS, Nuprl/MetaPRL.

GENERAL PURPOSE PROOF ASSISTANTS

Prehistory: de Bruijn's Automath Project Modern architecture: Robin Milner (1972) Stanford LCF (Logic for Computable Functions). Circa 1979 - Edinburgh's LCF - tactics, isolated trusted core, proof checker: HOL, Coq, Mizar, Isabelle, PVS, Nuprl/MetaPRL.

Most use a goal-driven derivation: the user starts from the goal and “decomposes” (refines) it down to axioms and/or established facts (top-down derivation). At every moment, a partial derivation is a tree with possible ungrounded leaves. It becomes complete when all leaves are ground.

HOL

Stands for (classical) Higher-Order Logic, uses predicate calculus with terms from typed Mike Gordon (Cambridge University), 1988, a direct descendant of Edinburgh LCF.

Current versions: (HOL88, HOL90, HOL98, HOL Light, HOL 4). Mathematics formalized in HOL: real analysis up to fundamental theorem of calculus, complex numbers up to fundamental theorem of algebra, weak form of the Prime Number Theorem, floating-point arithmetic, etc.

HOL-LIGHT

HOL Light was designed by John Harrison and Konrad Slind, runs on standard PCs, and supports both top-down and down-top derivations. It has been used in the Flyspeck project to machine-check Tom Hales's proof of the Kepler conjecture. Success so far: the Jordan Curve Theorem.

Coq

Coq, INRIA, is based on Coquand's Calculus of Inductive Constructions (1985), extension of Girard's polymorphic λ . Its main goal was specification and verification of programs. Coq's basic logic is intuitionistic, and it includes a mechanism for automatic generation of certified programs from proofs of their specifications

Coq

Coq is widely used for formalization of mathematics: real analysis, constructive category theory, elements of constructive geometry, group theory, domain theory, fundamental group theory. A recent success story: formalization and verification of a proof of the Four Color Theorem (1999/2004).

MIZAR

Non-interactive proof-checker, forward style from axioms to goals. Started in 1974 (Andrzej Trybulec) as software to support a working mathematician in preparing papers.

Logic: classical first-order, natural deduction.

Mathematics: Tarski-Grothendieck set theory.

MIZAR

Journal Formalized Mathematics (a computer assisted approach) established in 1990 and devoted solely to the formalizations of mathematics in Mizar. All papers are checked by the Mizar. They formalized the Jordan Curve Theorem.

Mizar Mathematical Library includes 926 articles written by 175 authors and 41525 theorems, 7838 definitions, 722 schemes, 6805 registrations, 5784 symbols, 1903 keywords.

ISABELLE

Isabelle (started in 1986, Larry Paulson, Cambridge University, and Tobias Nipkow, TU Munich), rather a logical framework (“generic proof assistant”), not tightly bound to one specific logic.

Meta-logic is intuitionistic higher-order logic with equality; different logical systems can be defined: HOL, FOL, ZF, HOL with Scott's Logic for Computable Functions (domain theory) added, small fragment of Martin-Lof's Type Theory (ITT), Barendregt's Lambda Cube, and others.

ISABELLE

Large theory library: elementary number theory (for example, Gauss's law of quadratic reciprocity), analysis (basic properties of limits, derivatives, and integrals), algebra (up to Sylow's theorem), and set theory (the relative consistency of the Axiom of Choice), the Prime Number Theorem.

PVS

Stands for Prototype Verification System, SRI International, commenced in 1990, intended for significant applications. PVS is a research prototype: it evolves and improves as the stress of real use exposes new requirements.

Based on simply typed classical higher-order logic extended with subtyping, dependent typing, and parametric theories which makes it somewhat closer to Coq and Nuprl. Mathematical library: calculus, domain theory, program semantics, graph theory, a very elaborate library of decision procedures used for hardware and software verification.

NUPRL

PRL = Proof Refinement Logic, 1973, Nu = a version indicator. NuPRL appeared around 1984, Robert Constable, Cornell, now versions 1-5.

Built around Martin-Lof's Type Theory (ITT), a higher-order intuitionistic system. Aimed at program specification and verification, has an impressive list of successes. Nuprl is also a direct descendant of Edinburgh LCF.

NUPRL

Formalized mathematical theories including but not limited to constructive real analysis, computational abstract algebra (multivariate polynomial arithmetic, unique factorization domains), extracting constructive content from classical proofs, automata theory, Turing machines, etc.

Some major protocol verification successes.

MORAL SO FAR

Proof assistants are considered safe, if they produce an elementary proof checked by the trusted core. Elaborate system of tactics (lemmas, rules) provide a comfortable level of flexibility and extendability.

Should be used in combination with other methods, e.g. model-checking.

FORMAL METHODS IN REAL LIFE

All proof assistants mentioned (but, perhaps, Mizar) have been targeting verification applications, all have impressive success records.

Hires of formal method experts by industry. Harrison (HOL light) is now Intel's senior engineer.

In programming languages the state of the art is almost at the point where an electronic appendix with machine-checked proofs accompanying papers is fast becoming the norm.

CONCLUSIONS

Computer-aided proofs are playing an increasingly prominent role.

Computers bring precision to proof building. Computer-verified proofs are more reliable than those verified by humans.

Proof assistants are sometimes the only tool capable of handling an increasing complexity beyond the capacity of any human being.

New layer of challenges in this area.

It takes a different set of skills to formalize a long proof than to find one.