Formal methods and verification: what have we achieved?

Pierre Wolper
The beginning

- **1967 – 1969**: Assertions, invariants, axiomatic semantics; emphasis on logic and proofs (*Floyd, Hoare*).
  - Small sequential programs.
  - Programs with simple specifications.
  - Proving programs is a worthy activity (Dijkstra).
  - Theorem provers as possible support.
A Turning Point

• **1975 – 1985**: Focus on concurrency, finite-state systems, decidability and the introduction of algorithmic methods.
  – Dynamic logic as a formalization of Hoare logic in the finite-state case (Pratt).
  – Temporal logic to specify ongoing behaviors (Pnueli).
  – Process algebras (Milner).
  – Synthesis from propositional temporal logic (Clarke-Emerson, Manna-Wolper).
  – Model-Checking (Clarke-Emerson, Sifakis-Queille).
The Algorithmic Explosion

• **1985 – 1995**: An explosion of methods to cope with the *state explosion problem*.
  – Automata theoretic methods (Vardi-Wolper)
  – Symbolic methods: coding sets of states by formulas (BDDs).
  – Partial-order methods: avoiding unnecessary states when modeling concurrency (Valmari, Godefroid-Wolper, Peled).
  – SAT based methods.
  – Abstraction, exploiting symmetry.
  – Verifying systems with sets of identical processes.
  – Proof assistants, expressive logics, process algebras.
The extension period

• **1995 – 2005:** Moving beyond finite-state systems and towards control systems.
  – Models of timed systems: timed automata.
  – Symbolic methods for infinite-state systems: arithmetic, queues, pushdown systems.
  – Hybrid systems: including continuous dynamics.
  – Probabilistic systems.
  – Using solvers for Satisfiability Modulo Theories (SMT).
Adapting to a Changing World

• **2005 – 2015:** What needs to be verified is changing fast.
  – Hardware is reaching very large sizes and is as intricate as earlier software systems.
  – Processors are multicore and handle shared memory differently (relaxed memory models).
  – Security is a crucial issue.
  – If the network fails everything stops.
  – Biological data and systems need to be analyzed.
  – Embedded systems are not what they use to be.
What have we achieved

• Academic respectability.
• Fairly large research community, dedicated conferences.
• Many Awards, including Turing Awards: Dijkstra, Floyd, Hoare, Milner, Pnueli, Clarke, Emerson, Sifakis.
• Activity in industrial research labs.
• Use in industry: hardware, hardware design tools, systems, embedded systems.
What have we achieved

• Significant body of theoretical knowledge, potentially usable in other areas.
• Interesting developments in data structures and algorithms.
• Impressive development of proof tools (interest much wider than verification).
• Effective bug finding tools applicable in a variety of areas:
  – Hardware design
  – Programs with data abstracted
  – Device drivers
What have we not achieved

• Eliminating bugs: no shortage of bugs in the foreseeable future.
• Writing specifications, which are bug free.
• Automating theorem proving in expressive logics.
• Exploring very large state-spaces and keeping up with the increase of system sizes:
  – The size of systems grows exponentially (Moore’s law),
  – The state space is exponential in the system size,
  – Keeping up with this double exponential is really hard!
The Future

• Ever better tools for analysing systems, even if the analysis is not exhaustive.
• Niche applications for which the cost is justified:
  – Space probes,
  – Network security, internet of things,
  – Other very critical applications
• Expanding use of verification developed techniques in other areas.
• Radically different approaches: mining big code repositories.
• Build systems that can tolerate faults in some of their parts.
• We can live with (some) bugs.
Can we live with bugs?

• We do manage to live with humans, which are
  – Erratic,
  – Unpredictable,
  – Subject to mood changes,
  – Make lots of mistakes (arithmetic, spelling, ...)
  – Prone to making major errors (world wars, economic crisis, failing countries, ...)

• What makes it possible is that
  – We have many mechanisms for correcting errors (individually or collectively), and that
  – We don’t all make the same mistakes. Diversity helps.