Formal Verification Systems

A Presentation on Formal Methods

Outline

- Error-Free Systems and System Design
- History of Formal Verification and Hardware Verification
- Current Trends in Formal Verification
- Theorem Proving
- Logic in Theorem Proving
- Automated vs Interactive Theorem Proving
- Symbolic Simulation and Model Checking
- Conclusion and Future Research

Outline

- **Error-Free Systems and System Design**
- History of Formal Verification and Hardware Verification
- Current Trends in Formal Verification
- Theorem Proving
- Logic in Theorem Proving
- Automated vs Interactive Theorem Proving
- Symbolic Simulation and Model Checking
- Conclusion and Future Research

Error-Free Systems

- Air France Flight 447 (2009) crashed due to incorrect airspeed readings.

Error-Free Systems

- Air France Flight 447 (2009) crashed due to incorrect airspeed readings.
- Safety-critical domains (medicine, transport, chemical plants) rely on error-free systems.
- System bugs can endanger lives or cause financial loss.

System Design and Verification

- System Design: Ensures the system exhibits the desired behavior.

System Design and Verification

- System Design: Ensures the system exhibits the desired behavior.
- System Analysis: Uses mathematics to model and verify properties.

System Design and Verification

- System Design: Ensures the system exhibits the desired behavior.
- System Analysis: Uses mathematics to model and verify properties.
- Testing and Simulation alone cannot guarantee correctness, which is why formal methods are needed

Formal Verification Methods

- Formal verification is a mathematical approach that ensures a system meets its specifications with certainty

History of Formal Verification

- Advocated by Dijkstra and Knuth.
- Initial lack of interest due to software patching.
- Gained traction in 1980s for digital hardware verification.

Why Hardware Verification?

- Hardware is more regular and less obscure than software.
- Hardware bugs are costlier—replacing silicon chips is expensive.
- Post-Intel floating point division bug, formal verification became widespread.

Current Trends in Formal Verification

- Used for verifying complex, unpredictable systems.
- High precision but time-consuming.
- Current approaches balance precision with efficiency by combining traditional testing with formal methods
- Major methods: Theorem Proving, Symbolic Simulation, Model Checking.

Outline

- Error-Free Systems and System Design
- History of Formal Verification and Hardware Verification
- Current Trends in Formal Verification
- **Theorem Proving**
- Logic in Theorem Proving
- Automated vs Interactive Theorem Proving
- Symbolic Simulation and Model Checking
- Conclusion and Future Research

Theorem Proving

Main Idea

- The system to be verified is modeled mathematically.
- Properties of interest are verified using theorem provers which rely on formal logics as as a modeling medium

Why Logic?

But why Logic and not other languages like Java,English?

Why Logic?

- Programming languages introduce ambiguity and multiple interpretations.
- Theorem proving requires a logical language with:
  - Precise, rigorous reasoning.
  - Well-defined syntax and semantics.

Logics Used in Theorem Proving

1. **Propositional Logic**
   - Deals with propositions (statements that can be true or false).
   - Uses Boolean operators: $\wedge, \vee, \neg, \Rightarrow, \Leftrightarrow$.
   - Decidable – formulas can be automatically verified.
   - Limitation: Lacks expressiveness for complex systems.

Logics Used in Theorem Proving

2.First-Order Logic (FOL) Extends propositional logic with:

- Quantifiers: $\forall$ (for all), $\exists$ (there exists).
- Predicates: Functions that return Boolean values.
- Allows declaration of constants, function names, and free variables.

Drawback: Semi-decidable – some proofs require human intervention.

Logics Used in Theorem Proving

3.Higher-Order Logic (HOL)

- Most expressive logic form.
- Allows quantification over functions and sets.
- Suitable for systems with continuous and unpredictable elements.
- Requires mathematical modeling in a closed form.

Types of Theorem Proving

Based on user intervention:

- Automated Theorem Proving (ATP): Minimal user involvement.
- Interactive Theorem Proving (ITP): Requires human input to guide the proof process.

Automated vs. Interactive Theorem Proving

Feature	Automated Theorem Proving (ATP)	Interactive Theorem Proving (ITP)
Logic Used	Propositional or first-order logic.	Higher-order logic for infinite domains.
User Involvement	Fully automated, minimal human input.	Requires human guidance for proof steps.
Techniques Used	Uses SAT solvers, SMT solvers, BDDs, DPLL methods.	Uses LCF-style provers with functional programming (ML, Coq).

Outline

- Error-Free Systems and System Design
- History of Formal Verification and Hardware Verification
- Current Trends in Formal Verification
- Theorem Proving
- Logic in Theorem Proving
- Automated vs Interactive Theorem Proving
- **Symbolic Simulation and Model Checking**
- Conclusion and Future Research

Symbolic Simulation

Symbolic simulation bridges the gap between traditional simulation/testing and formal verification.

Main Idea:

- Uses **symbols (variables)** instead of actual values.
- Simultaneously considers **multiple executions**.
- Reduces the number of test cases, making **exhaustive simulation feasible**.

Model Checking

Definition: Model checking is an **automated verification technique** used for reactive systems. It verifies properties of finite-state models using temporal logic.

Key Components:

- **System Model** – Finite-state representation of the system.
- **Specification** – Properties expressed in temporal logic.
- **Verification Algorithm** – Exhaustively checks all states.

Advantages:

- Fully automatic verification process.
- Provides counterexample if a property fails.

Outline

- Error-Free Systems and System Design
- History of Formal Verification and Hardware Verification
- Current Trends in Formal Verification
- Theorem Proving
- Logic in Theorem Proving
- Automated vs Interactive Theorem Proving
- Symbolic Simulation and Model Checking
- **Conclusion and Future Research**

Conclusion

Formal verification ensures **precise system analysis**, essential for **safety-critical systems**, but it requires **significant time and effort**.

Conclusion

Each method has trade-offs: **theorem proving** is powerful but manual, **symbolic simulation** is automatic but limited in scope, and **model checking** is efficient but suffers from state-space explosion.

Future Research Directions

While formal methods are now applied beyond computing (e.g., **physics, biology, economics**), **industrial adoption remains limited** due to **complex tools, time constraints, and lack of trained engineers**.

Future Research Directions

Solutions include **better training, industry regulations**, and improving tool usability.

References

[1] O. Hasan and S. Tahar, "Formal Verification Methods," in *Formal Techniques for Safety-Critical Systems*, IGI Global, 2015, pp. 1-40. Available: `https://www.igi-global.com/chapter/formal-verification-methods/112414`

Thank You!

Questions?