Model Check Concurrent Systems

Hao Zheng
Dept. of Computer Science & Eng.
University South Florida
Verifying Current Systems

- Simulation samples a set of input vectors and observes outputs.
- Coverage becomes less sufficient when systems become larger.
  - Systems states grow exponentially in the number of state variables.
  - State transitions grow exponentially in the number of system states.
- Dramatic simulation failures:
  - Intel Pentinum processor.
  - Ariane 5 rocket explosion.
Deduction Reasoning

• Advantages:
  – Guarantees correctness of systems.
  – Can be applied to infinite systems.

• Disadvantages:
  – Proves systems using axioms and proof rules.
  – Requires strong mathematic background.
  – Mainly done by hand (cannot be fully automated).
  – Mostly used in safety critical applications.
Temporal Logic Model Checking

• Model checking is an automatic verification technique for finite state concurrent systems.
• Developed independently by Clarke, Emerson, and Sistla and by Queille and Sifakis in early 1980’s.
• Specifications are written in temporal logic, CTL, LTL, etc.
• Verification procedure is an exhaustive search of the state space of the design.
Advantages of Model Checking

• Exhaustiveness (compared to simulation).
• No proofs.
• Fast (compared to other rigorous methods such as theorem proving).
• Diagnostic counter-examples.
• No problem with partial specifications.
• Logics can easily express many concurrency properties.
Main Disadvantage

State Explosion Problem:
• Too many current processes
• Data Paths

Much progress has been made on this problem recently!
• Symbolic model checking
• Partial order reduction
• Abstraction
• Symmetry reduction
Reactive Systems

• Interact with environment.
• Never terminates.
• Need to model environment as well.
• A state is value of variables at a time instant.
• State transition is caused by actions in systems.
• Computation is infinite sequences of states connected by state transitions.
Representations of Concurrent Systems

- VHDL, Verilog, SystemC, UML, StateChart, etc.
- Need a unifying formalism.
- First order logic
  - Synchronous and asynchronous
- Petri nets
  - Asynchronous
- Automata
- They are translated to state transition graphs.
Microwave Oven Example

State-transition graph describes system evolving over time.

No concurrency in simple examples.
Temporal Logic

- The oven does not **heat up** until the **door is closed**.
- Not **heat_up** holds until **door_closed**
- \((\neg \text{heat\_up}) \cup \text{door\_closed}\)
Basic Temporal Operators

The symbol $f$ and $g$ are atomic propositions, e.g. “Device Enabled”.

- $Fp$ - $f$ holds sometime in the future.
- $Gp$ - $f$ holds globally in the future.
- $Xp$ - $f$ holds next time.
- $p U q$ - $f$ holds until $g$ holds.
Model Checking Problem

Let $M$ be a labeled state-transition graph (Kripke structure).

Let $f$ be the specification in temporal logic.

Find all states $s$ of $M$ such that $M, s \models f$.

Efficient Algorithms: CE81, CES83
A Model Checking System

Preprocessor → Specification

Model Checker

State Transition Graph
$10^4$ to $10^5$ states

True or Counterexamples
Breakthrough!


Model Checker SMV

Now able to handle much larger examples!!
(designs can have as many as $10^{120}$ states).
Approaches to State Explosion

• **Binary Decision Diagrams**
  – Represent state transition systems concisely.

• **Partial order reduction**
  – Reduce the number of states to be enumerated.

• Many techniques for alleviating state explosion:
  – **Abstraction**.
  – **Compositional reasoning**.
  – **Symmetry**.
  – **Cone of influence reduction**.
  – **Semantic minimization**.
Model Checker Performance

- Model checkers today can routinely handle systems with between 100 and 1000 state variables.

- Systems with $10^{120}$ reachable states have been checked. (Compare approx. $10^{78}$ atoms in universe.)

- By using appropriate abstraction techniques, systems with an essentially unlimited number of states can be checked.
Model Checking Systems

• There are many successful examples of the use of model checking in hardware and protocol verification.

• The fact that industry (INTEL, IBM, MOTOROLA) is starting to use model checking is encouraging.

• Below are some well-known model checkers.
Temporal Logic Model Checkers

- The first two model checkers were **EMC** (Clarke, Emerson, Sistla) and **Caesar** (Queille, Sifakis).

- **SMV** is the first model checker to use **BDDs**.

- **Spin** uses the **partial order reduction** to reduce the state explosion problem.

- **Verus** and **Kronos** check properties of **real-time systems**.

- **HyTech** is designed for reasoning about **hybrid systems**.
Notable Examples - HDLC

- A **High-level Data Link Controller** was being designed at AT&T in Madrid in 1996.

- Researchers at Bell Labs offered to check some properties of the design using the **FormalCheck verifier**.

- Within five hours, **six properties were specified and five were verified**.

- The sixth property failed, uncovering a **bug** that would have **reduced throughput** or caused **lost transmissions**!
Notable Examples
PowerPC 620 Microprocessor

• Richard Raimi used Motorola’s Verdict model checker to debug a hardware laboratory failure.

• Initial silicon of the PowerPC 620 microprocessor crashed during boot of an operating system.

• In a matter of seconds, Verdict found a BIU deadlock causing the failure.
Bounded Model-Checking

• Problem: How to compute set of reachable states?
  – Fixpoint computation is too expensive.

• Idea: Restrict search to states that are reachable from initial state within fixed number $n$ of transitions

• Implemented by unwinding programs into propositional formulas.

• Applying SAT solver to the formulas
  – If there exists a assignment to the variables that makes formulas satisfied, then there is a bug!
Bounded Model-Checking (cont.)

• Advantages
  – Avoid expensive fixpoint computation.
  – Number $n$ can be adjusted to available computation power.
  – Find bugs if systems terminate quickly.

• Disadvantages
  – Runtime grows with bound.
  – Might be too small for large examples.
Topics to be Covered

• Propositional/Predicate logic
• Computational Logic
• Explicit model checking
• Binary decision diagram
• Symbolic model checking
• Abstraction
• Compositional verification/assume-guarantee reasoning
• SAT/Bounded model checking
• Inference
• Timing verification