내장형 소프트웨어를 위한 모델 검증 기법

Tae-Jin Kim, Moonzoo Kim

Software Testing & Verification Group
KAIST

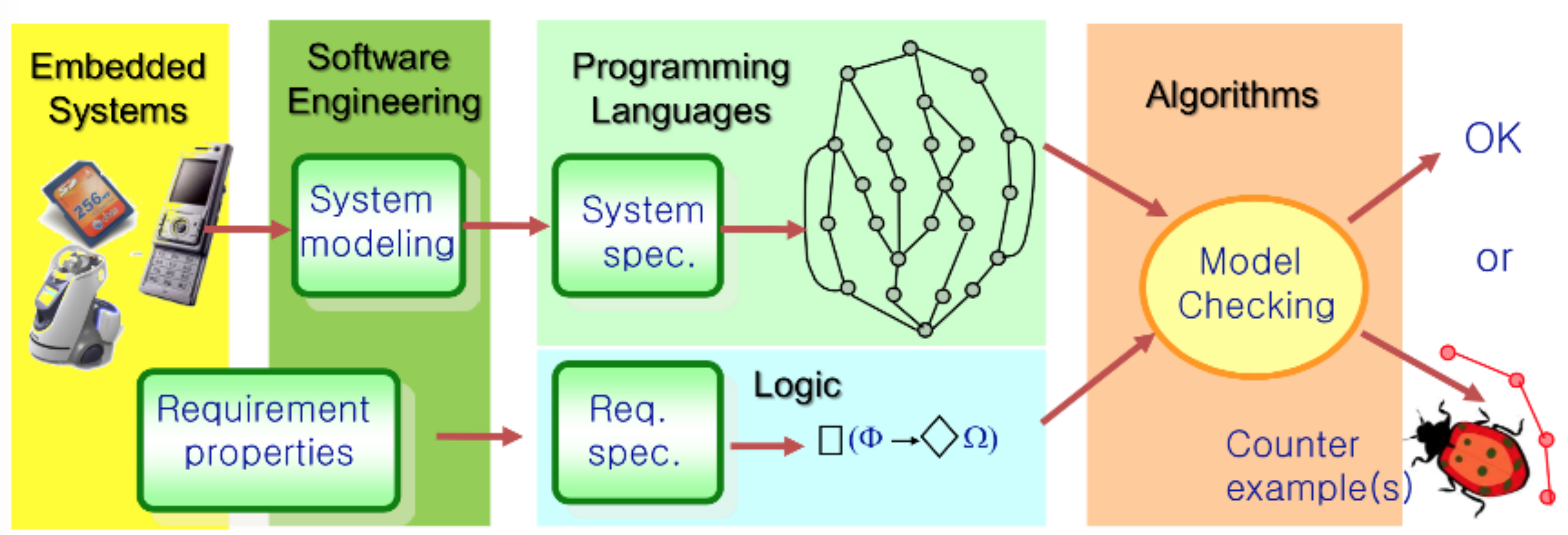
Motivation

- Software errors on embedded software systems may cause catastrophic accidents
- It is necessary for software engineers to assure the reliability of the target system
- Model checking techniques can be good solutions as these assure the correctness of the target program model, or detect error execution scenarios

Explosion of Ariane 5

Process of Model Checking

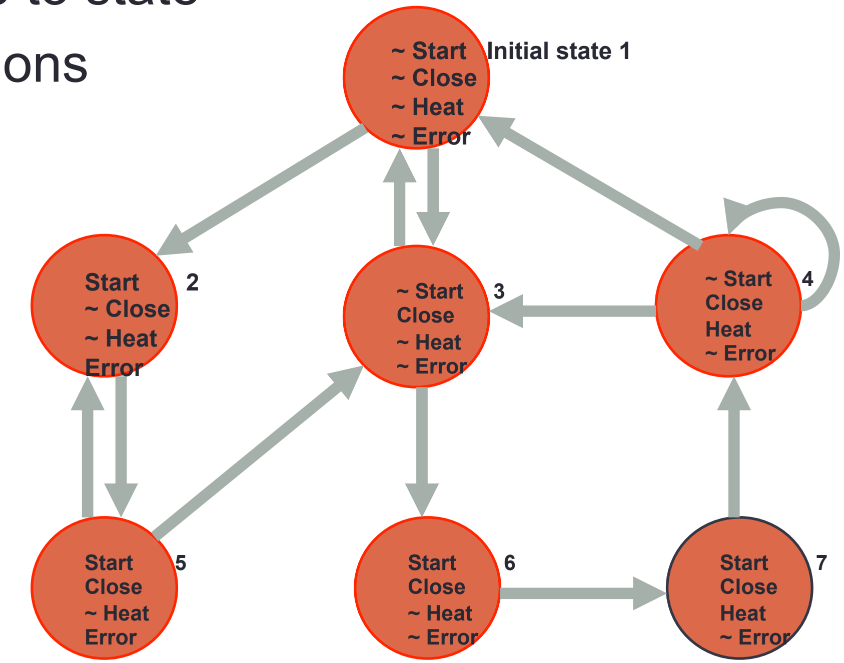
- System Modeling
  - Construct a model that can capture the properties for verification
  - Depending on characteristics of system (i.e. async/synchronous circuit)
- Specification
  - Need to describe properties that the model must satisfy
    e.g. “For All situation, Always No Deadlock”
- Verification
  - Check that the system satisfies the specification (Satisfiability/Counter example)



System & Requirement Specification

- System specification
  - Let model M = (S, R, L) be a Kripke structure to represent the behavior of a system
    - S is the finite set of states
    - R is a transition relation from state to state
    - L is a function that labels propositions

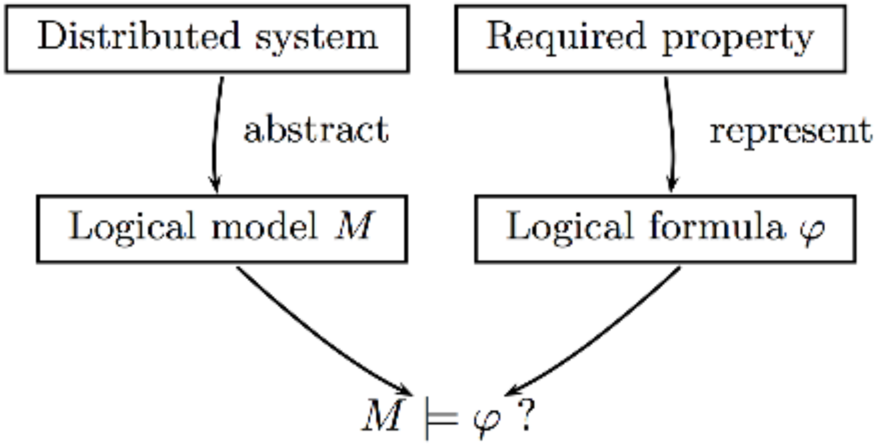
- Requirement specification
  - Use Temporal logic formula
    - Describing sequences of transitions between states
    - “For all paths,
      Always Error occurs then no Heat”
      can describe as “AG(Error→¬Heat)”



Kripke Structure of a Microwave

Model Checking

- Given a model M and a temporal logic formula φ, model checking is the problem of verifying whether or not φ is true in M (written as M ⊨ φ)
  - Find all states s of M, which satisfies φ. That is, {s| M,s ⊨ φ}



Symbolic Model Checking

- Explicit model checking techniques represent each state and transition relations explicitly
  - Each state is represented as a valuation of state variables
  - These techniques cost too much memory space
    - Hash table based explicit model checker generates 2^{40} states for an array of 5 character elements
    - State Explosion Problem

- Symbolic model checking techniques represent a set of states and the transition relations as Boolean logic formulas, and checks it
  - Use Ordered Binary Decision Diagrams (OBDD) to reduce the memory space for state space

Future Plan

- Research on efficient symbolic model checking techniques for real-world systems
    - Study about partial order reduction which is a technique for reducing the size of the state space
    - Apply the techniques (OBDD, partial order reduction), which reduce the size of state space, on real-world systems and figure out how much the memory space is actually reduced
    - Find the new technique to solve the state explosion problem on model checking

Thank you