Theorem Proving versus Model Checking

EECS 4315

www.eecs.yorku.ca/course/4315/

Comparison

In theorem proving, programs are typically verified method-by-method. Given a method and its contract, a theorem prover transforms the precondition while symbolically executing the method. Then, it checks whether the transformed precondition is a model of the postcondition (i.e., it implies the postcondition).

In model checking, programs are usually verified by means of test scenarios. A model checker takes the program and test scenarios as input and exhaustively searches for possible violations. The difference of test scenarios compared to test cases is that they can include arbitrary values (e.g., a boolean value or a positive integer), which are all considered during model checking.

Comparison

"In model-checking, you describe an abstracted version of your system and you can automatically check some properties. Your model has to be small enough (in term of number of states) to be processed by the tool, and the class of formulas you can express may be limited. Typically, you can check safety properties (such as assert statement in your code).

On the other hand, using a theorem prover (I think "proof assistant" is a better term), you can work on more accurate representations of your system and express any properties, but most proofs have to be done manually which requires time and expertise."

Comparison

"In model-checking, you describe an abstracted version of your system and you can automatically check some properties. Your model has to be small enough (in term of number of states) to be processed by the tool, and the class of formulas you can express may be limited. Typically, you can check safety properties (such as assert statement in your code).

On the other hand, using a theorem prover (I think "proof assistant" is a better term), you can work on more accurate representations of your system and express any properties, but most proofs have to be done manually which requires time and expertise."

Comparison

Model checking and theorem proving go about different ways to answer the question.

Model checking, roughly, tries to use brute force to answer the question and requires no human interaction in doing so. You could imagine it feeding every possible input to every process, choosing every possible interleaving of messages and, for every state reachable in such a manner, checking whether the bad thing happens.

In theorem proving, you try to provide the rationale of why things can't go wrong in form of theorems. However, you also have to convince the theorem prover that your reasoning is sound. So first you need to understand what methods of reasoning you are using precisely, and you also need to somewhat understand the way of how the prover "ticks" and what kinds of reasoning steps it can perform automatically.

Comparison

Model checking and theorem proving go about different ways to answer the question.

Model checking, roughly, tries to use brute force to answer the question and requires no human interaction in doing so. You could imagine it feeding every possible input to every process, choosing every possible interleaving of messages and, for every state reachable in such a manner, checking whether the bad thing happens.

In theorem proving, you try to provide the rationale of why things can't go wrong in form of theorems. However, you also have to convince the theorem prover that your reasoning is sound. So first you need to understand what methods of reasoning you are using precisely, and you also need to somewhat understand the way of how the prover "ticks" and what kinds of reasoning steps it can perform automatically.

Model Checking

Advantages

- Automatic.

Limitations

- Can only handle "small" models (and models may become huge due to the state space explosion problem).
- Can only check properties expressed in "simple" logics such as LTL and CTL.

Theorem Proving

Advantages

- Can handle code of arbitrary size.
- Can check any property.

Limitations

- Labour intensive.
- Can only be used by experts.

Finding Bugs . . .

. . . is still a challenge. Being familiar with

- testing,
- theorem proving and
- model checking

is essential for tomorrow's software engineer.

Final Exam

When: Thursday April 13, 9:00-11:00

Where: Accolade East, room 006

What: all the material covered in the course with emphasis on the material that has not yet been tested in quizzes (the textbook contains useful exercises).

Note: you may bring one letter sized piece of paper with notes on a single side; relevant definitions will be provided in the exam.