Formal Verification

Paul Wild

Wednesday 16th October, 2024

Tutorial procedure

Homework

- 50% of your grade comes from homework exercises
- exercise sheets will appear regularly
- submission via StudOn, usually until before the next tutorial

Tutorial procedure

Homework

- 50% of your grade comes from homework exercises
- exercise sheets will appear regularly
- submission via StudOn, usually until before the next tutorial

Class

- homework presentation, comparison of solutions, discussion of problems
- we will experiment with the tools during class
- active participation required

Model Checking and Temporal Logics

Model-based verification techniques

- describe a *system* and its behaviour in a mathematically precise and unambiguous manner

Model Checking and Temporal Logics

Model-based verification techniques

- describe a *system* and its behaviour in a mathematically precise and unambiguous manner
- use algorithms to explore all possible states of the models

Model Checking and Temporal Logics

Model-based verification techniques

- describe a *system* and its behaviour in a mathematically precise and unambiguous manner
- use algorithms to explore all possible states of the models
- **model checking:** exhaustive search

Model Checking and Temporal Logics

Model-based verification techniques

- describe a *system* and its behaviour in a mathematically precise and unambiguous manner
- use algorithms to explore all possible states of the models
- **model checking:** exhaustive search
- **simulation:** experiments with a restrictive set of scenarios

Model Checking and Temporal Logics

Model-based verification techniques

- describe a *system* and its behaviour in a mathematically precise and unambiguous manner
- use algorithms to explore all possible states of the models
- **model checking:** exhaustive search
- **simulation:** experiments with a restrictive set of scenarios
- **testing:** experiments on a "real" implementation of the model

Model Checking and Temporal Logics

Model-based verification techniques

- describe a *system* and its behaviour in a mathematically precise and unambiguous manner
- use algorithms to explore all possible states of the models
- **model checking:** exhaustive search
- **simulation:** experiments with a restrictive set of scenarios
- **testing:** experiments on a "real" implementation of the model

Model Checking and Temporal Logics

Model-based verification techniques

- describe a *system* and its behaviour in a mathematically precise and unambiguous manner
- use algorithms to explore all possible states of the models
- **model checking:** exhaustive search
- **simulation:** experiments with a restrictive set of scenarios
- **testing:** experiments on a "real" implementation of the model

The Spin model checker

- provides a modelling language to describe systems consisting of multiple processes running concurrently, while communicating and using shared resources

Model Checking and Temporal Logics

Model-based verification techniques

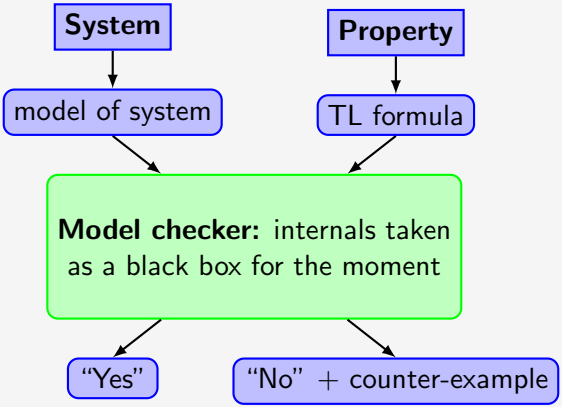
- describe a *system* and its behaviour in a mathematically precise and unambiguous manner
- use algorithms to explore all possible states of the models
- **model checking:** exhaustive search
- **simulation:** experiments with a restrictive set of scenarios
- **testing:** experiments on a "real" implementation of the model

The Spin model checker

- provides a modelling language to describe systems consisting of multiple processes running concurrently, while communicating and using shared resources
- has many tools for model checking, simulation and testing

Model Checking and Temporal Logics

Model Checking



Installing Spin

Spin

- Official website and documentation: `https://spinroot.com/spin/Man/README.html`
- Source code and precompiled binaries: `https://github.com/nimble-code/Spin/tags`

Installation, checking and troubleshooting

- Follow the OS-specific instructions on the next slide.
- Go to `Examples/` and follow the README instructions to run some checks.

Installing Spin

Windows

- Follow these instructions: `https://blog.nathanv.me/posts/spin-windows`
- Download MinGW here instead: `https://sourceforge.net/projects/mingw`
- We will not be making use of the GUI components, so you can ignore that part.
- Add both the locations of Spin and GCC to your path.
- You may also want to install VSCode and its C/C++ and Promela extensions.

Linux/Mac

- Many package managers already have Spin available.
- If not, download the GitHub release and either compile yourself or use the binary.
- Add `spin` to your path for convenience.