

Principles of Symmetry Reduction for Verification and Testing

Sümmeetriareduktsioonide põhimõtted verifitseerimises ja testis

Juhan Ernits

Dept. of Comp. Sci / Inst. of Cybernetics
Tallinn University of Technology

Pesa @ Põlva 25.01.2008

Overview

- ▶ Motivation
- ▶ General overview of state space reduction techniques
- ▶ Symmetry reduction techniques
 - ▶ Equivalent markings in Petri Nets [Jensen 85]
 - ▶ Scalar sets in model checking [Ip and Dill 93]
 - ▶ Heap canonization [Iosif 2002]
 - ▶ Extraction of channel diagrams [Miller and Donaldson 2007]
 - ▶ State graph isomorphism [Rensink 2006, Veanes et al 2007]
- ▶ Summary och ToDo

Motivation

- ▶ Exploit symmetries for reducing the required amount of storage in automatic exploration of models of systems.

Automated Analysis and Verification

- ▶ There is a variety of automated techniques available.
- ▶ We look at different techniques used in:
 - ▶ *Explicit state* model checking
 - ▶ Model-based testing
- ▶ Model checking relies on a *formal model of the system* and model-based testing on a *formal model of the specification* that can be automatically analysed.

Modelling Formalisms

We look at three modelling formalisms that are created for modelling software systems with automation in mind:

- ▶ *Timed Automata*, the modelling formalism of the real-time model checker Uppaal and the implementation of scalar sets
- ▶ *Model programs* that use C# and a custom library called *NModel* for model-based testing
- ▶ *Graphs*, as in Groove
- ▶ ...

State Space Explosion

- ▶ The techniques of *model checking* and *model-based testing* involve traversing the state space of the *model*.
- ▶ There may be many states, for example, there are:
 - ▶ an estimated 10^{81} atoms in the Universe
 - ▶ more than $10^{1200000}$ possible states in a processor with 4 MB of cache
- ▶ Even abstract models can easily have state spaces that can not be enumerated with reasonable amount of time and resources.
- ▶ We need to reduce the number of states to be traversed to be successful

State Space Reduction Techniques

- ▶ Partial order reductions
- ▶ **Symmetry reductions**
- ▶ Symbolic methods
- ▶ Compositional methods
- ▶ Incomplete techniques
 - ▶ hash compaction
 - ▶ bitstate hashing, Bloom filters

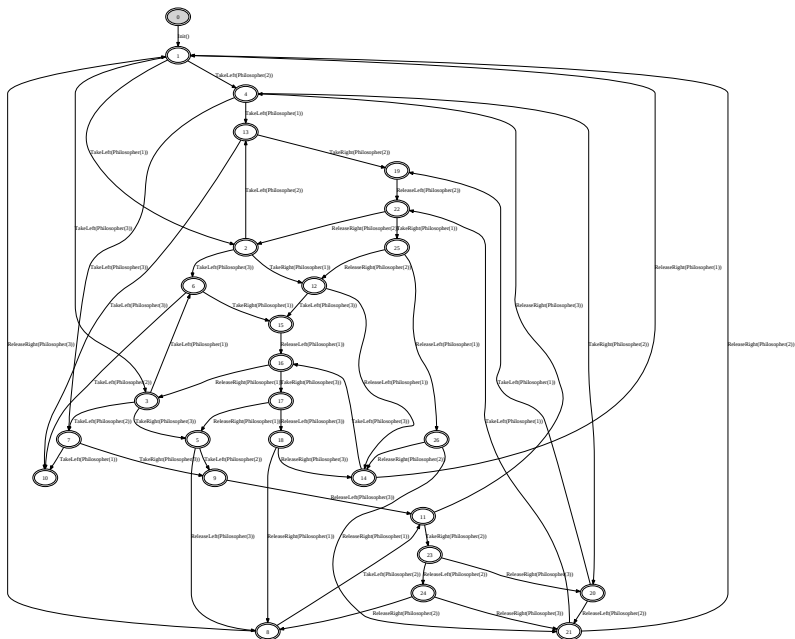
Scalar Sets

- ▶ Scalar set is an integer subrange with restricted operations.
- ▶ Let us look at an example!

Model Programs

- ▶ C# programs that use the API provided by NModel
- ▶ Have built in *abstract data types*: Sets, Maps, Bags
- ▶ Can contain objects with arbitrary structure
- ▶ Model programs use *non-determinism* on action selection and action argument selection level
- ▶ Model program state is the configuration of all the variables between actions.

Model Programs: State Space of 3 Dining Philosophers



Program State as a Graph

Model Program Viewer

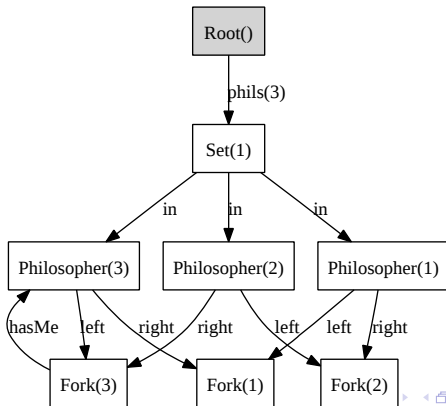
File

PhilsWith

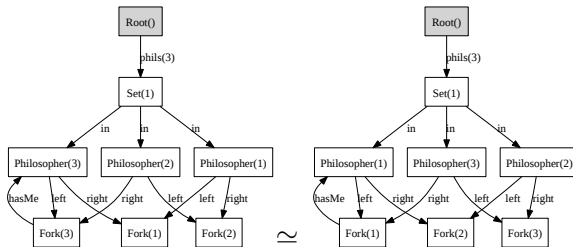
Variables in state 1

PhilsWithObjects	
[Domain Map	Map("Fork", 3, "Philosopher", 3)
hasMe	Map(Fork(2), null, Fork(3), null, Fork(1), null)
left	Map(Philosopher(1), Fork(1), Philosopher(3), Fork(3), Philosopher(2), Fork(2))
mode	ControlMode("Running")
phils	Set(Philosopher(1), Philosopher(3), Philosopher(2))
right	Map(Philosopher(1), Fork(2), Philosopher(3), Fork(1), Philosopher(2), Fork(3))
state	Map(Philosopher(1), State("Thinking"), Philosopher(3), State("Thinking"), Philosopher(2), State("Thinking"))

PhilsWithObjects



State Graph Isomorphism



- ▶ Works well in a highly symmetric example
- ▶ Is applicable for more symmetries than *scalar sets*
- ▶ Is an important approach as there is an independent implementation by Arend Rensink in GROOVE

Summary

- ▶ Symmetry reduction provides an exact abstraction
- ▶ Of all currently known methods, state graph isomorphism is the most general, but is also quite resource intensive
- ▶ To Do
 - ▶ The statically symmetric topology could be remembered also for the isomorphism checking
 - ▶ Experiment with improving the efficiency of isomorphism checking. One idea: use multiset discrimination based approach by Paige and Tarjan as suggested by Henglein

Thank you for your attention!
Questions?