Reactive Systems: Model Checking Tools

Timo Latvala

April 28, 2004

Model Checking Tools

There is a wide variety of Model checking tools available. Selecting the right tool for the job is sometimes hard. Here are some personal opinions/suggestions for different model checking tools.

SPIN

- Modelling formalism: Promela language, systems can be seen as a set of synchronized extended finite state machines
- Model Checkers: on-the-fly LTL, safety through assertions, LTL to Büchi translation
- Other features: partial order reductions and bit-state hashing can be combined with LTL model checking, model slicing
- Comments: Very fast state space generation, fast partial order reduction algorithm, hash table stored in physical memory (instead of on a file on hard-disk), nice xspin GUI
- Suggested uses: Modeling of communication protocols, LTL model checker

NuSMV2

- Modelling formalism: SMV input language, a simple circuit description language
- Model Checkers: BDD based CTL and LTL model checkers (under fairness), bounded model checking with LTL (including past operators).
- Other features: Deadlock checking, computing the number of reachable states, simulation
- Comments: A strong BDD based CTL model checker (a SMV rewrite), a reasonable bounded LTL model checker
- Suggested uses: verifying digital circuits (or systems easily modeled as circuits), CTL under fairness model checking, bounded model checking

MARIA

- Modelling formalism: Algebraic Petri nets (including P/T-nets)
- Model Checkers: on-the-fly LTL model checking under (strong and weak) fairness, safety
- Other features: extensive support for structured datatypes, parallel safety model checker
- Comments: useful for systems with complex data manipulations, uses disk to manage larger statespaces
- Suggested uses: systems with lots of fairness constraints, as a back-end for programming languages (datatype support eases this tremendously)

PROD tool

- Modelling formalism: High-level Petri nets with integer tokens (including P/T-nets)
- Model Checkers: on-the-fly LTL model checking with partial order reductions, safety, livelock detection
- Other features: off-line CTL model checker, stubborn sets and sleep sets
- Comments: very strong partial order reductions available (once you know which flags to use), weak LTL to Büchi translation algorithm, partial order reduction algorithms are internally implemented on low-level Petri nets
- Suggested uses: systems which can be conveniently modeled with low-level Petri nets, where partial order reductions are important

MURφ tool

- Modelling formalism: guarded command rules (contains some subsets of P/T-nets + much more)
- Model Checkers: safety model checking
- Other features: symmetry reductions, parallel model checking, hash compaction
- Comments: symmetry reductions, slower than SPIN on some models
- Suggested uses: symmetric systems, parallel model checking

Other Model Checking Tools

Here is a short list of model checking tools, which are somewhat outside of the scope of this course. I give also the suggested application domain.

- Petri net tools: INA, Lola, PEP, Design/CPN, ...
- Uppaal & Kronos: Two tools for model checking timed systems
- Caesar Aldebaran (CADP): A set of model checking tools based on LTSs
- Java Pathfinder 2: Model checker for Java programs
- Bandera: Java abstraction and slicing system, with model checking back-ends
- Slam: A (Microsoft) tool for model checking C programs
- Model Checking Kit: A collection of model checkers in one tool

Related courses

- T-79.146 Logic in Computer Science: Special Topics I
- T-79.154 Logic in Computer Science: Special Topics II
- T-79.185 Verification
- T-79.193 Formal Description Techniques for Concurrent Systems