TACAS 20th Anniversary

Benchmarks and Benchmarking: The Model Checking Contest

Hubert Garavel (Inria Grenoble – LIG and Saarland U.)

joint work with

Fabrice Kordon (UPMC – LIP6)

Lom Messan Hillah (UPOND – LIP6)

Alban Linard (Inria Saclay – LSV)

Beyond tool papers…

- What tool papers do well:
  - Report about latest software advances
  - Provide a quick summary of tool functionalities

- What is more difficult to achieve:
  - Guarantee the veracity of assertions about tools (e.g., functionality, performance, user-friendliness)
  - Enable reproducibility of experiments
  - Fairly compare different tools/algorithms providing similar functionalities

Current trend: Tool contests

- **HWMCC** *Hardware Model Checking Competition*
  - Armin Biere et al.
  - http://fmv.jku.at/hwmcc13
- **RERS** *Rigorous Examination of Reactive Systems*
  - Bernhard Steffen
  - http://rers-challenge.org
- **SV-COMP** *International Competition on Software Verification*
  - Dirk Beier — satellite event of TACAS 2014
  - http://www.sosy-lab.org/~dbeyer/Publications/2014-TACAS.Status_Report_on_Software_Verification.pdf
- **MCC** *Model Checking Contest*
  - Fabrice Kordon et al. — satellite event of Petri Nets 2014
  - http://mcc.lip6.fr

A multifaceted problem…

- Verification of real systems faces many issues:
  - Logics and decision procedures
  - Complex data structures
  - Large fragments of sequential code
  - Concurrency: message-passing, shared-memory
  - Quantitative time
  - Performance and reliability aspects
- So far, these aspects are addressed 1 by 1, or 2 by 2
- Ultimately, they should be addressed together

Benchmarks

Example 1: the VLTS suite

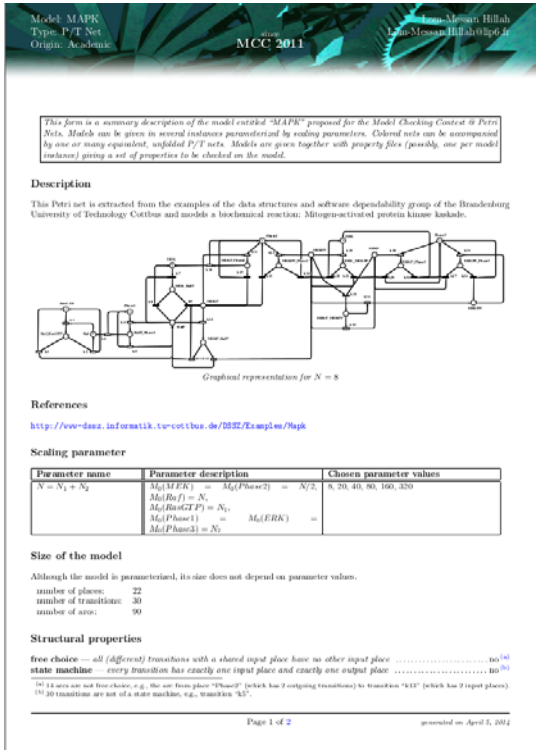
- **VLTS** *Very Large Transition Systems*
  - Stefan Blom (CWI) and Hubert Garavel (INRIA) —2003
  - http://cadp.inria.fr/resources/vlts
  - A collection of 40 explicit Labelled Transition Systems
  - Increasing sizes from 300 states to 34 million states
  - Derived from industrial case studies with concurrency
- Seems to address a real need:
  - No publication about VLTS
  - No advertisement of any kind
  - Yet used and cited in 38 scientific publications

Example 2: the MCC challenge

- **MCC** *Model Checking Contest*
  - Yearly event since 2011 – 4th edition in 2014
  - Launched by Fabrice Kordon and colleagues
  - http://mcc.lip6.fr
  - Oriented towards highly-concurrent systems
- Two main features of MCC:
  - Benchmarks: *Call for models*
  - Benchmarking: *Call for tools*

MCC models

- Petri net models (encoded in PNML format)
  - Place-transition and/or colored (with unfoldings)
  - Possible scaling parameters (initial tokens or colors)
  - Safe or not (multiple arcs and tokens)
- A growing set of diverse models
  - 2011: **7**, 2012: **12**, 2013: **9**, 2014: **15**
  - Diverse origins: many universities
  - Diverse types: hardware, software, manufacturing, bioinformatics, etc.
  - Diverse sizes: scaling parameters


Model: MAPK
Type: P/T Net
Origin: Academic
MCC 2011

This form is a summary description of the model entitled "MAPK" proposed for the Model Checking Contest @ Petri Nets. Models can be given in several instances parameterized by scaling parameters. Colored nets can be accompanied by one or many equivalent, unfolded P/T nets. Models are given together with property files (possibly, one per model instance) giving a set of properties to be checked on the model.

Description

This Petri net is extracted from the examples of the data structures and software dependability group of the Brandenburg University of Technology Cottbus and models a biochemical reaction: Mitogen-activated protein kinase kaskade.

Graphical representation for $N = 8$

References

http://www-dssz.informatik.tu-cottbus.de/DSSZ/Examples/Mapk

Scaling parameter

Parameter name	Parameter description	Chosen parameter values
$N = N_1 + N_2$	$M_0(MEK) = M_0(Phase2) = N/2$, $M_0(Raf) = N$, $M_0(RasGTP) = N_2$, $M_0(Phase1) = M_0(ERK) = M_0(Phase3) = N_1$	8, 20, 40, 80, 160, 320

Size of the model

Although the model is parameterized, its size does not depend on parameter values.

number of places:	22
number of transitions:	30
number of arcs:	90

Structural properties

free choice — *all (different) transitions with a shared input place have no other input place* ..... no [a]
state machine — *every transition has exactly one input place and exactly one output place* ..... no [b]

[a] 14 arcs are not free choice, e.g., the arc from place "Phase2" (which has 2 outgoing transitions) to transition "k13" (which has 2 input places).
[b] 30 transitions are not of a state machine, e.g., transition "k5".

Page 1 of 2 — generated on April 3, 2014

MCC properties

- Structural properties
  - Net size, free choice, state machine, marked graph, etc.
- Behavioral properties
  - Marking graph size, safe, live, bounds, deadlocks, etc.
- Temporal logic formulas
  - Manually written by the authors of some models
  - Randomly generated by the MCC team
    - Atomic propositions: place cardinality, transition fireability, etc.
    - Connectors: reachability formulas, LTL formulas, CTL formulas
  - Generating "meaningful" formulas is difficult

Benchmarking

The BenchKit technology

- How to measure maximal memory and CPU usage?
- For sequential applications:
  - Uppaal's **Memtime** tool
  - *(we have patches for Memtime, e.g., 64-bit support)*
- For concurrent applications (processes / threads)
  - **BenchKit** tool http://benchkit.cosyverif.org
  - Based on virtual machine technology – multi OS
  - Suitable for clusters and many core machines
  - Evaluates: user time, average CPU time, maximal memory usage, and their evolution in time

Tool benchmarking at MCC 2013

- 12 competing tools (submitted as VMs)
- 24 models, 255 model instances
- 4335 examinations per tool (instances × properties)
- Computation of results using BenchKit:
  - Hardware: 3 academic clusters totalling 104 cores
  - Running time: 84 days and 6 hours
  - Execution traces: 1.89 GB of text + csv data
- Analysis of results:
  - Automatic tools required to process such huge data
  - Manual handling of "paradoxical" situations

Conclusion

Conclusion

- Need for benchmarks and benchmarking
- Two concurrency-oriented benchmarks:
  - VLTS *Very Large Transition Systems*
  - MCC *Model Checking Contest*
- Generic results, reusable for other studies:
  - 40 large, documented Labelled Transition Systems
  - 43 large, documented Petri nets
  - Enhanced Memtime tool
  - BenchKit technology
  - Random generator of temporal logic formulas