qcomp.org

Competing with Probabilities: Challenges and Outcomes of QComp

Arnd Hartmanns
University of Twente

based on joint work with Carlos E. Budde, E. Moritz Hahn, Christian Hensel, Sebastian Junges, Michaela Klauck, Joachim Klein, Jan Křetínský, David Parker, Tim Quatmann, Enno Ruijters, Marcel Steinmetz, Andrea Turrini, and Zhen Zhang

QComp: A Quantitative Competition

"Friendly competition": no ranking

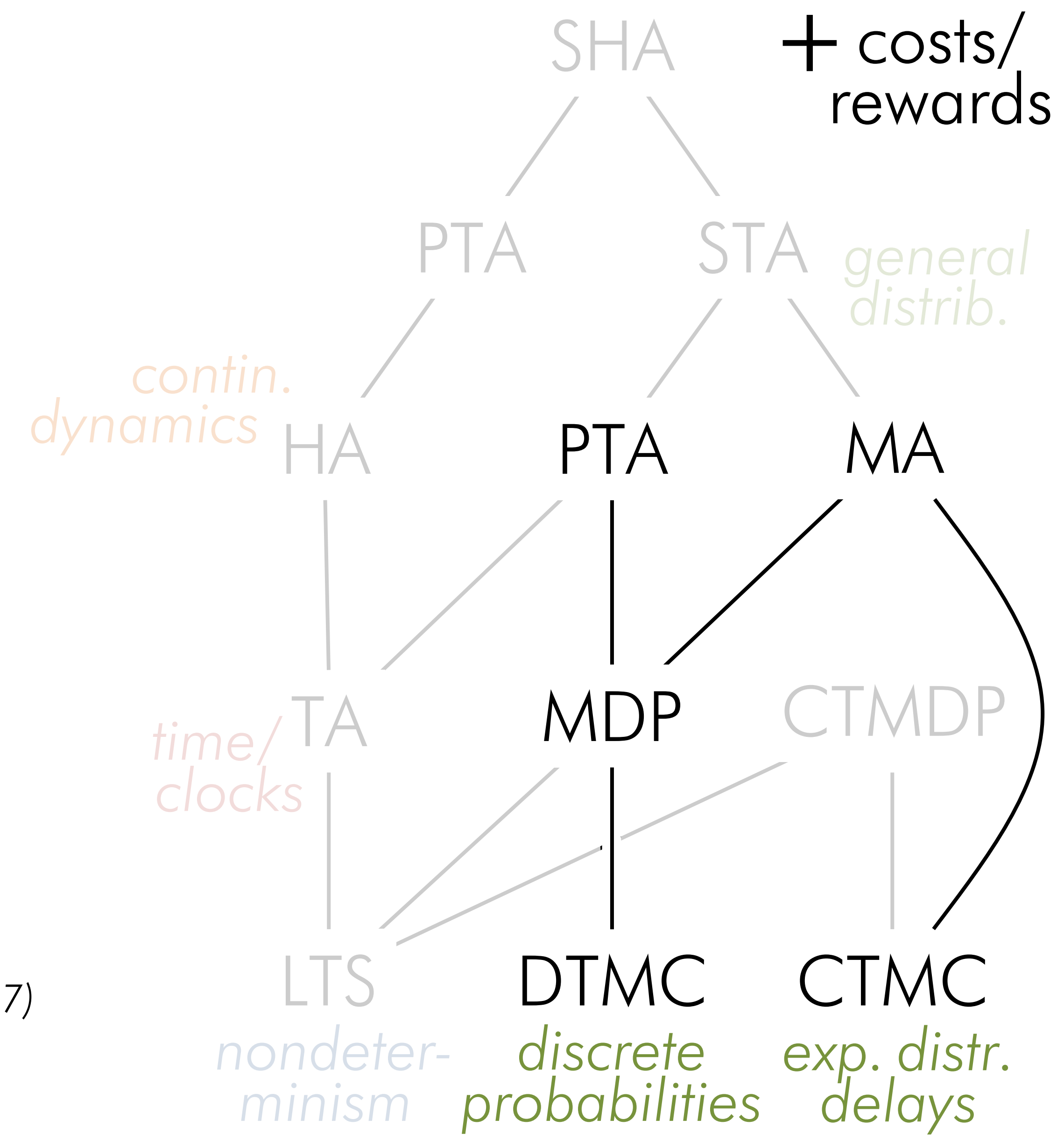
Semantic formalisms:

DTMC, CTMC, MDP, MA, PTA

Modelling languages:

GreatSPN	*stochastic Petri nets*
PPDDL	*planning domains*
PRISM	*general, low-level*
…	*and several others*
+ JANI	*model exchange format*

(`jani-spec.org`, *TACAS'17*)



QComp: A Quantitative Competition

Properties to check:

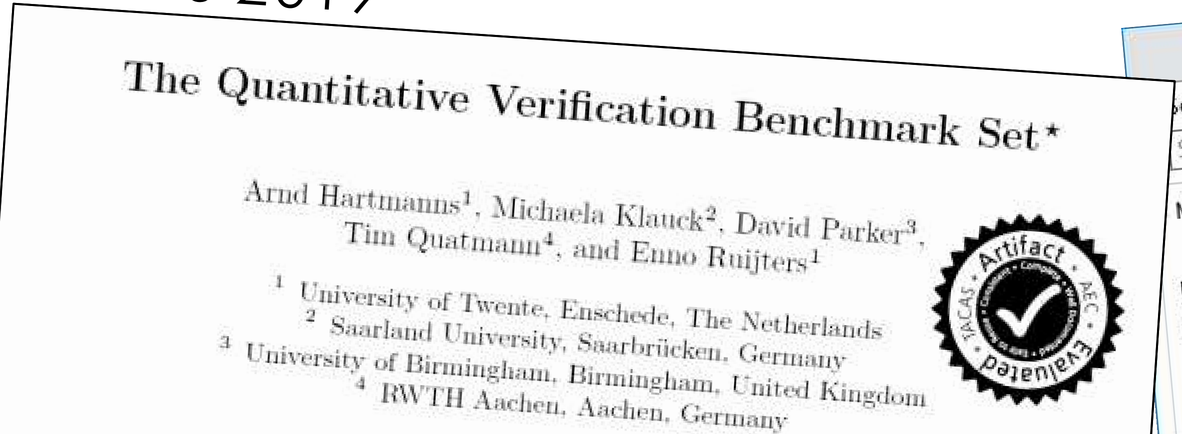
reachability probability	$\mathbb{P}(\diamond\, G)$	unbounded, time-, reward-bounded
expected reward	$\mathbb{E}(\mathrm{cost} \rightarrow G)$	
steady-state probability	$\mathbb{S}(G)$	

Benchmarks from the Quantitative Verification Benchmark Set

all QVBS entries must have a JANI version

QVBS

TACAS 2019

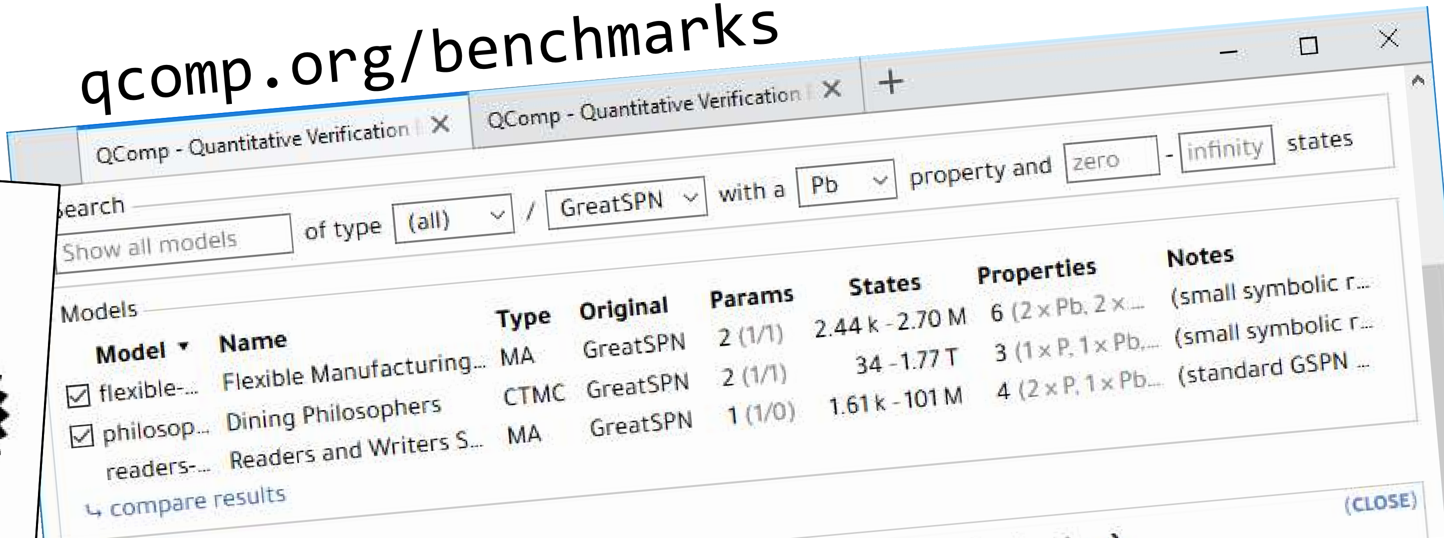


The Quantitative Verification Benchmark Set*

Arnd Hartmanns[1], Michaela Klauck[2], David Parker[3], Tim Quatmann[4], and Enno Ruijters[1]

[1] University of Twente, Enschede, The Netherlands
[2] Saarland University, Saarbrücken, Germany
[3] University of Birmingham, Birmingham, United Kingdom
[4] RWTH Aachen, Aachen, Germany

qcomp.org/benchmarks



The Competitors: Algorithms

Probabilistic Model Checking PMC

= numeric algorithm on full state space

– limited by state space explosion

+ ϵ-correct results: $|v - \bar{v}|/v \leq \epsilon$

(unknown) true value ↑ computed result ↑

Statistical Model Checking SMC

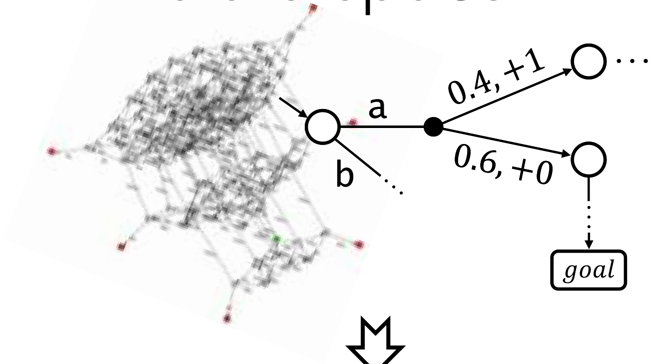
PMC:

formal model

```
process P() {
 alt {
 :: stop {= fail = true =}
 :: send palt {
    :95: {= done = true =}
    : 5: reset; P()
} } }
```

⇩

state space



⇩

precise results

$\mathbb{P}_{\text{min}}(\diamond\, a) = 0.2035$

$\mathbb{P}_{\text{max}}(\diamond\, a \wedge b) = 0.89$

$\mathbb{E}_{\text{min}}(\#s \mid b) = 12.5$

Probabilistic Model Checking PMC

= numeric algorithm on full state space

− limited by state space explosion

+ ϵ-correct results: $|v - \bar{v}|/v \leq \epsilon$

(unknown) true value   computed result

Statistical Model Checking SMC

= formal Monte Carlo simulation

+ constant memory usage

− rare events, nondeterminism

PAC guarantee: $\mathbb{P}(|v - \hat{v}| > \epsilon) < 1 - \delta$

estimate   confidence, e.g. 95 %

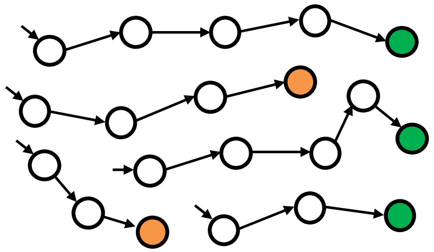
SMC:

formal model

```
process P() {
 alt {
 :: stop {= fail = true =}
 :: send palt {
    :95: {= done = true =}
    : 5: reset; P()
} } }
```

⇩

sample runs



⇩

estimated results

$\mathbb{P}_{\text{min}}(\diamond\, a) \approx 0.2$

$\mathbb{P}_{\text{max}}(\diamond\, a \wedge b) \approx 0.9$

$\mathbb{E}_{\text{min}}(\#s \mid b) \approx 12$

Probabilistic Model Checking PMC

= numeric algorithm on full state space

− limited by state space explosion

+ ϵ-correct results: $|v - \bar{v}|/v \leq \epsilon$

(unknown) true value ↑ ↑ computed result

Statistical Model Checking SMC

= formal Monte Carlo simulation

+ constant memory usage

− rare events, nondeterminism

PAC guarantee: $\mathbb{P}(|v - \hat{v}| > \epsilon) < 1 - \delta$

↑ estimate ↑ confidence, e.g. 95 %

+ Hybrid Approaches

reinforcement learning

deep learning

truncation

partial exploration, guided by simulation

probabilistic planning

Challenges to Correctness

Challenges to Correctness

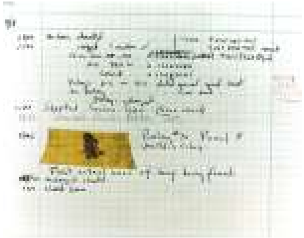
1. Bugs in algorithms:

   the algorithm itself is incorrect

e.g. sound value iteration:
small bug in helper method
pseudocode in original paper,
wrong in 1 of 79 test models

2. Bugs in implementations:

   the algorithm is correct, but the implementation is not



Acceptable?

Solutions: verify the algorithm with a theorem prover
correct-by-construction implementations
program verification

Isabelle

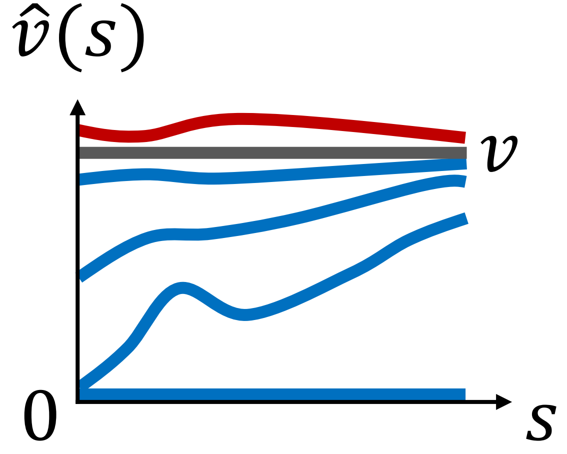
not specific to the quantitative setting

Challenges to Correctness

3. Unsound algorithms:

often but not always deliver ϵ-correct result

→ value iteration and derived algorithms with one-sided approximation of the fixpoint only



Solutions: interval iteration, optimistic value iteration, BRTDP, …

4. Floating-point implementations:

results unpredictably affected by rounding, cancellation, …

Solutions: exact rational arithmetic, safe rounding

does not scale

new

specific to probabilistic model checking

Challenges to Correctness

5. The statistical error in SMC:

up 5% of the results may be totally wrong, and that's okay

Recall PAC guarantee: $\mathbb{P}(|v - \hat{v}| > \epsilon) < 1 - \boxed{\delta}$

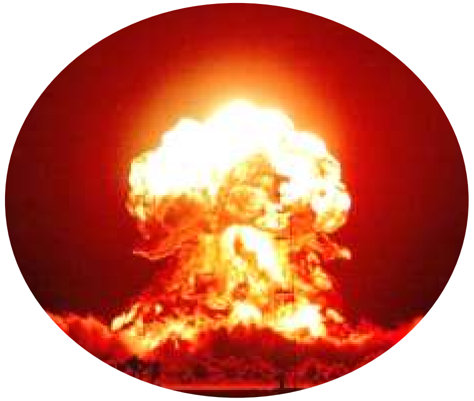
true value · estimate · error · confidence e.g. 95%

→ unavoidable in a statistical approach,
quantifiable (user-selectable confidence level)

How can we deal with these challenges in a tool competition?

Correct Quantitative Competitions

Option N: *Disqualify any tool that produces just a single (ϵ-)incorrect result and publicly shame its authors*



Consequences: All SMC tools disqualified
No unsound algorithms allowed
Floating-point implementations out

maybe that's a good idea?

→ only STORM and PRISM remain, using their limited *exact* engines

not representative of today's quantitative verification tools

Option SMC:

Use statistical test on statistical tools to assure confidence δ is adhered to

evaluation time explosion

The QComp 2020 Approach

QComp 2020: Tracks

Option QC20: *Use different tracks for different guarantees*

correct: must match true rational value where known $\epsilon = 0$

floating-point correct: must use algorithm that gives exact result, but may use floating-point arithmetic $\epsilon = 10^{-14}$

ϵ-correct: unconditionally require $|v - \bar{v}|/v \leq \epsilon$ $\epsilon = 10^{-6}$

probably ϵ-correct: require $\mathbb{P}(|v - \hat{v}| > \epsilon) < 1 - \delta$ from algorithm, but we do not check this statistically $\epsilon = 5 \cdot 10^{-2}$

often ϵ-correct: should ensure $|v - \bar{v}|/v \leq \epsilon$, but may sometimes be wrong (also with 10′ bound) $\epsilon = 10^{-3}$

QComp 2020: Tools

ePMC	PMC	*modular tool, focus on LTL*	ISCAS
mcsta	PMC	*disk-based, focus on correctness*	Twente
Prism	PMC	*the original probabilistic model checker*	Birmingh.
Storm	PMC	*has all the algorithms and languages*	Aachen
DFTRES	SMC	*dynamic fault tree rare event simulator*	Twente
modes	SMC	*rare events and nondeterminism*	Twente
MFPL	hybrid	*probabilistic planning using LRTDP*	Saarland
PET	hybrid	*the partial exploration tool*	Munich
Stamina	hybrid	*truncation for infinite-state CTMC*	Utah

QComp 2020: Tools

Tool	Galileo	GreatSPN	Jani	Modest	PGCL	PPDDL	Prism	DTMC			CTMC				MDP			MA				PTA		
								P	Pr	E	P	Pt	E	S	P	Pr	E	P	Pt	E	S	P	Pt	E
DFTRES	✓		✓					+			+	✓	+	✓				+	✓	+	✓			
ePMC			✓				✓	✓		✓	✓	✓	✓		✓		✓							
mcsta			✓	✓				✓	✓	✓	✓	✓	✓	+	✓	✓	✓	✓	✓	✓	+	✓	✓	✓
modes			✓	✓				✓	✓	✓	✓	✓	✓	+	✓	✓	✓	✓	✓	✓	+	✓	✓	✓
MFPL			✓	✓											+		+							
Prism							✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓					✓	✓	✓
PET							✓	✓			✓				✓									
Stamina							+					+												
Storm	✓	✓	✓		✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓

Tool capabilities; + marks additions since 2019

QComp 2020: Tools

Tool participation in the different tracks:

track	DFTRES	ePMC	mcsta	modes	MFPL	Prism	PET	Stamina	Storm
correct	—	—	—	—	—	—	—	—	✓
floating-p.	—	—	✓	—	—	—	—	—	✓
ε-correct	—	—	✓	—	—	✓	✓	—	✓
probably ε	✓	—	✓	✓	—	✓	✓	✓	✓
often ε	✓	✓	✓	✓	✓	✓	✓	✓	✓
often ε (10')	✓	—	✓	✓	✓	—	✓	✓	✓

→ specialised tools and generalists:
focus on specific algorithm vs. toolset

								Properties																
								DTMC			CTMC				MDP			MA				PTA		
Tool	Galileo	GreatSPN	Jani	Modest	PGCL	PPDDL	Prism	P	Pr	E	P	Pt	E	S	P	Pr	E	P	Pt	E	S	P	Pt	E
DFTRES	✓		✓					+			+	✓	+	✓				+	✓	+	✓			
ePMC			✓				✓	✓		✓	✓	✓	✓		✓		✓							
mcsta			✓	✓				✓	✓	✓	✓	✓	✓	+	✓	✓	✓	✓	✓	✓	+	✓	✓	✓
modes			✓	✓				✓	✓	✓	✓	✓	✓	+	✓	✓	✓	✓	✓	✓	+	✓	✓	✓
MFPL			✓	✓											+		+							
Prism							✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓					✓	✓	✓
PET							✓	✓			✓				✓									
Stamina							+					+												
Storm	✓	✓	✓		✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓

QComp 2020: Tuning the Tools

Some tools provide many options and algorithms.

Which to use to win the competition?

Default configuration: evaluate tool like a non-expert user

Specific tuning per instance: showcase the tool’s abilities

QComp 2020:

default = configuration *per track, modelling formalism, and property type* recommended by authors <u>today</u>

specific = aggressively tuned *per instance*; not used by all tools

(tool defaults may be historical)

QComp 2020: Tuning the Tools

New in STORM:

automatic selection of analysis configuration
based on syntactic aspects of the benchmark

…using a decision tree learned from the QComp benchmarks

→ default/specific distinction now pointless

Q: do we compare tools or algorithms?

Pragmatic solution for QComp 2020:

STORM + STORM-STATIC

↑ automatic ↑ as in QComp 2019

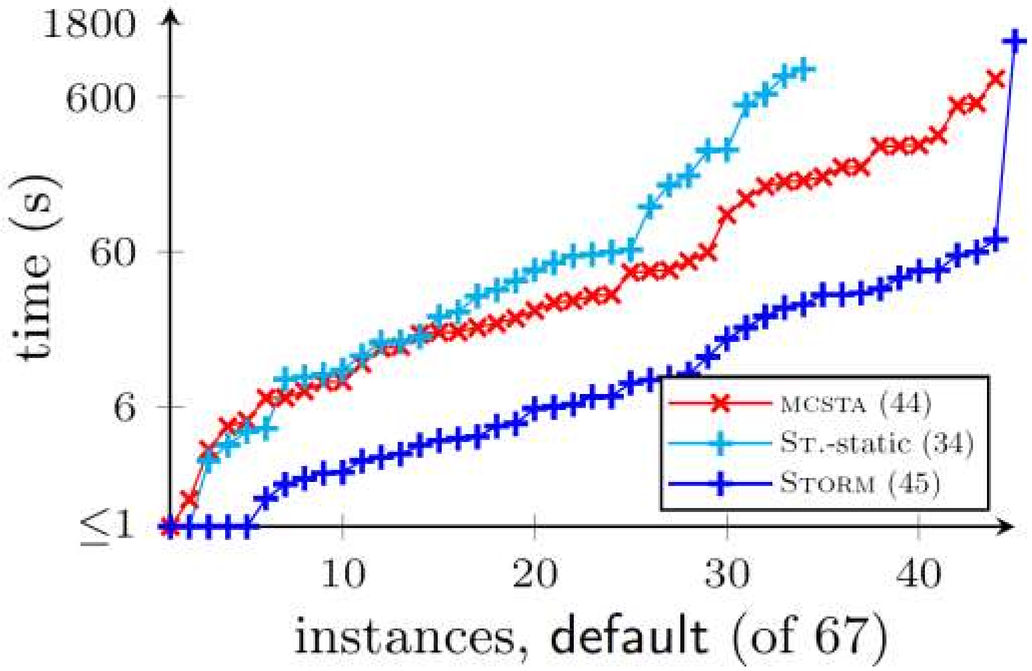
QComp 2020: The Results

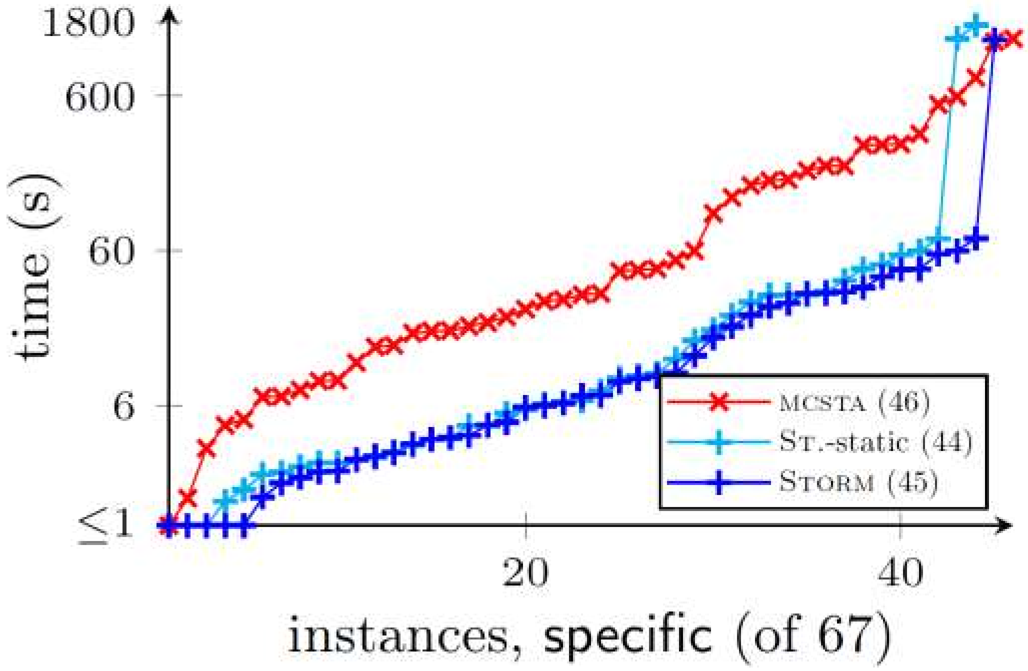
QComp 2020: Results

100 <u>*benchmark instances*</u>, from the QVBS

⟨model, parameters, property⟩

Quantile plots for overall comparison: **restricted to intersection**





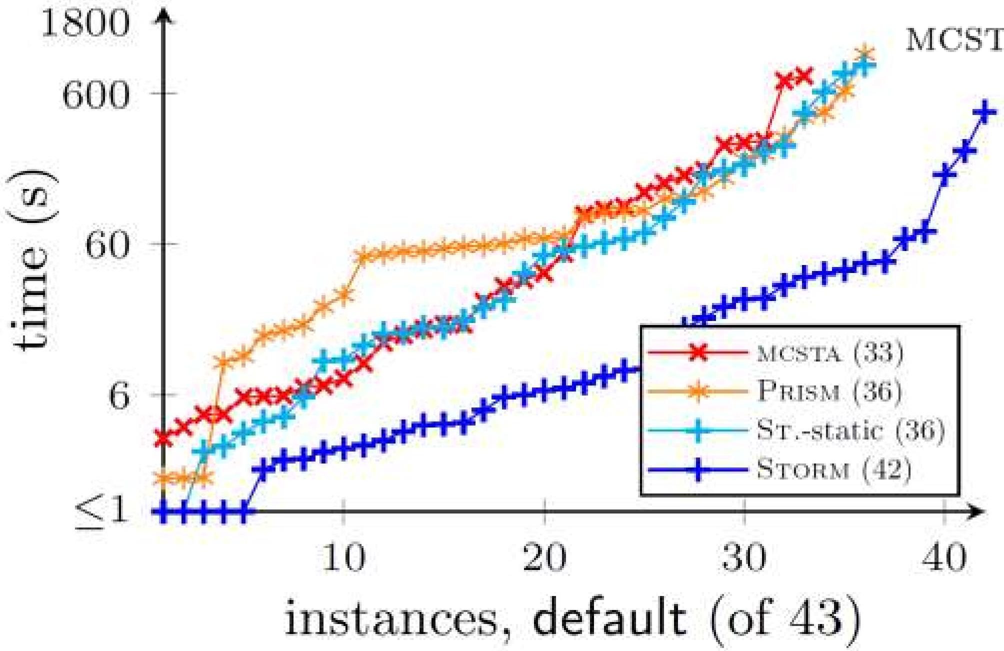
floating-point correct track

→ observe STORM vs. STORM-STATIC

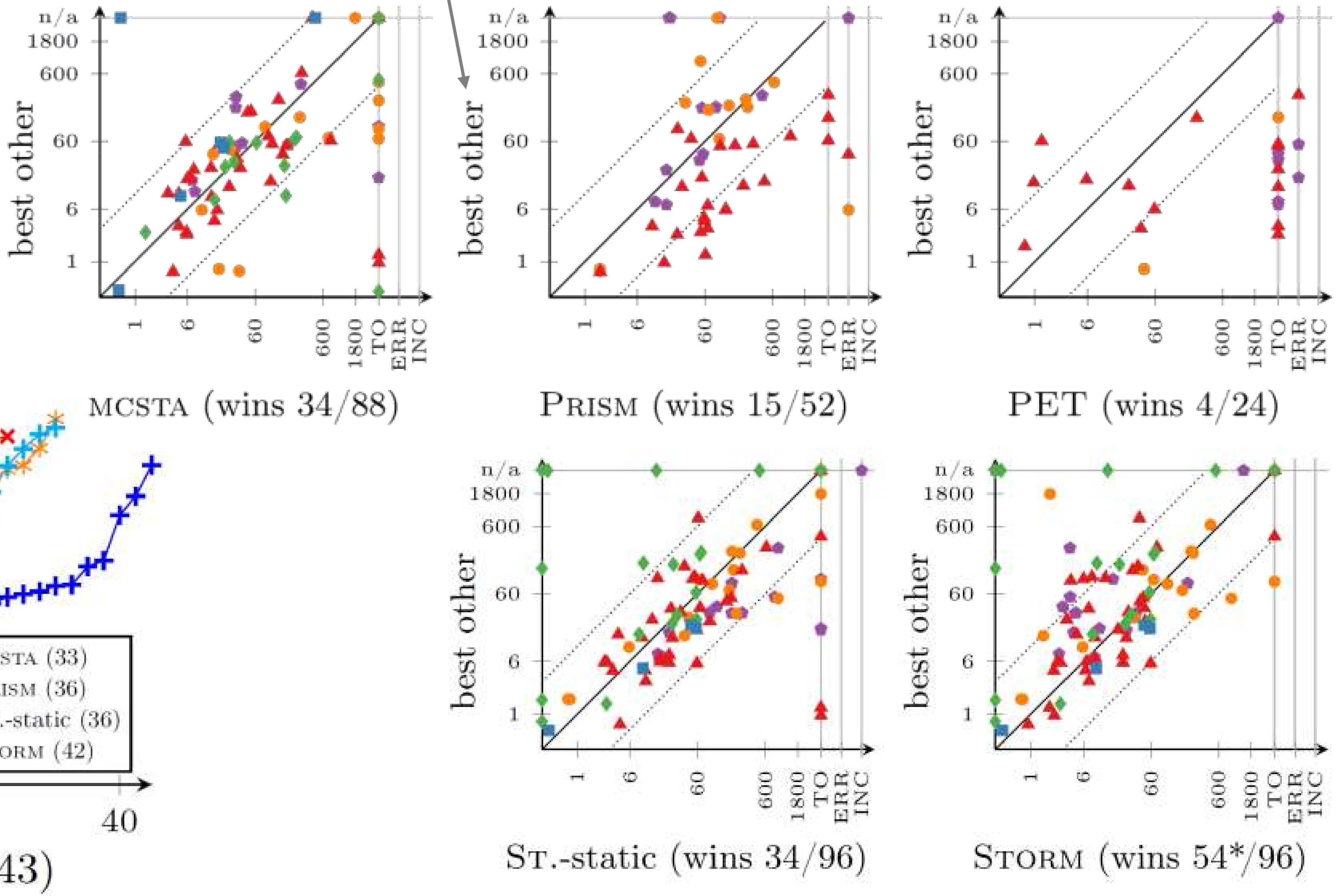
QComp 2020: Results

ϵ-correct track:

PMC tools + PET



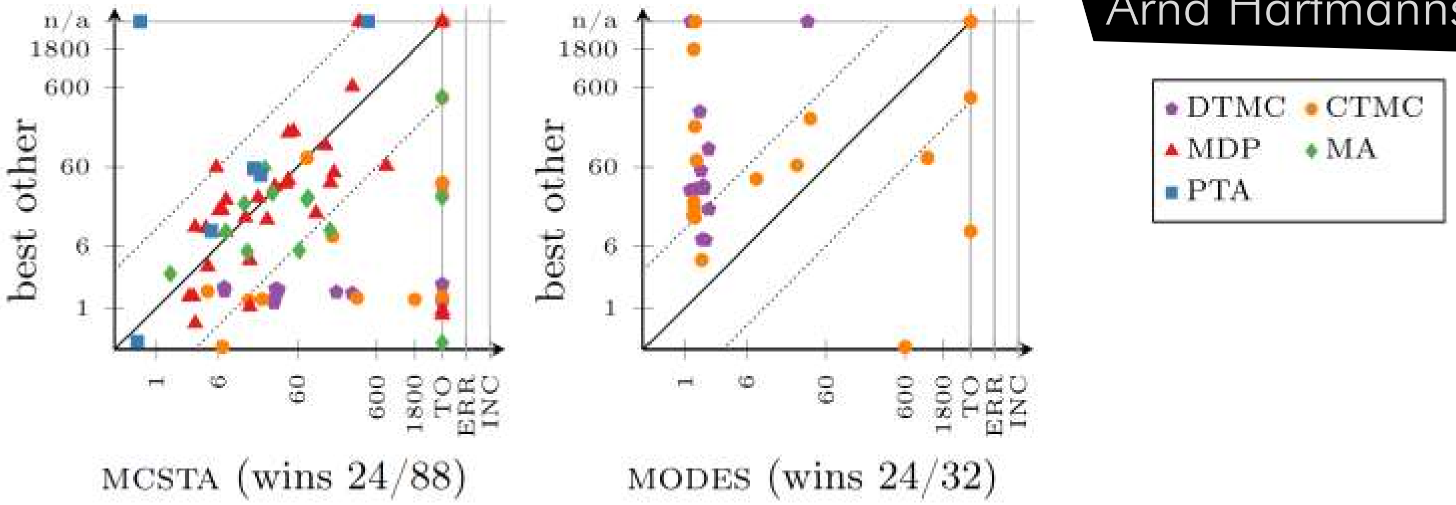
excl. STORM

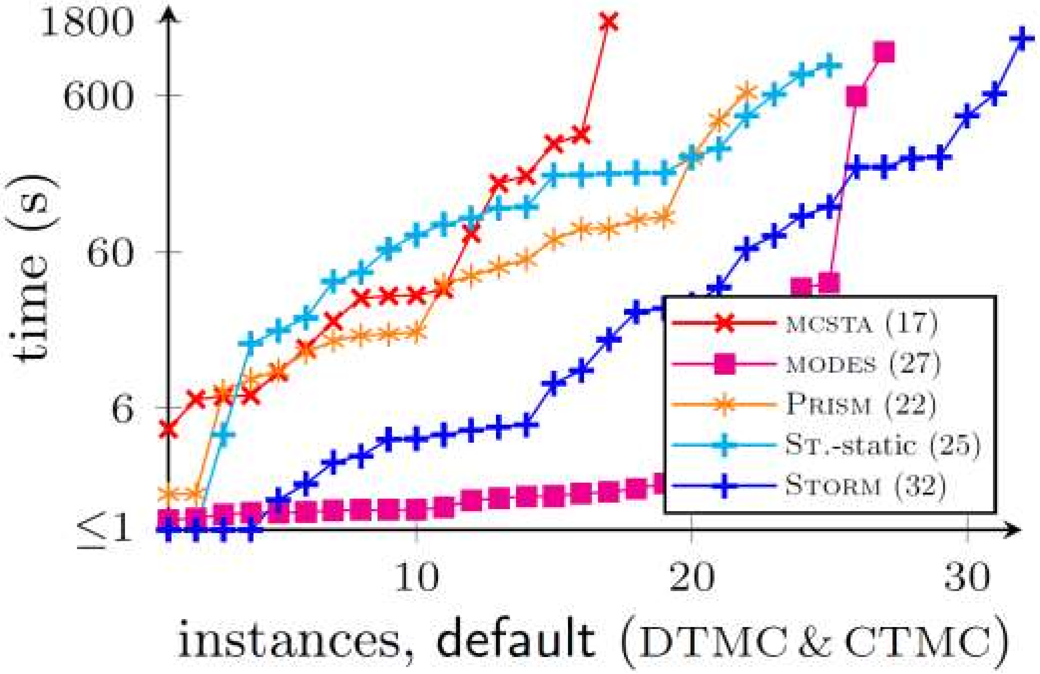


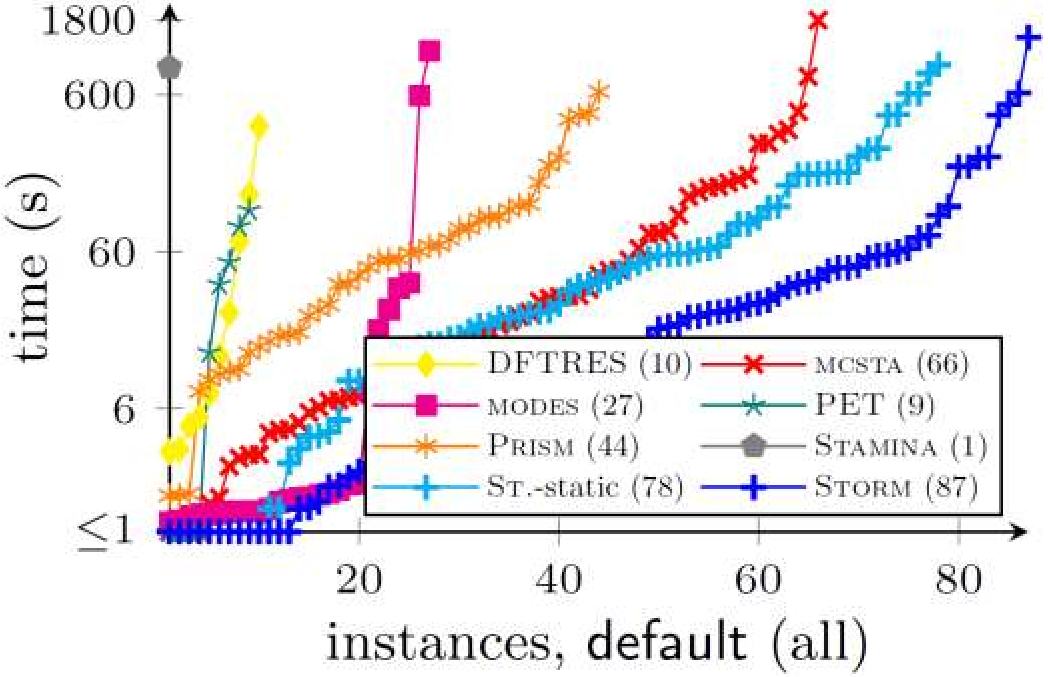
→ scatter plots show more details

QComp 2020: Results

Probably ϵ-correct track:
showcase for statistical model checkers





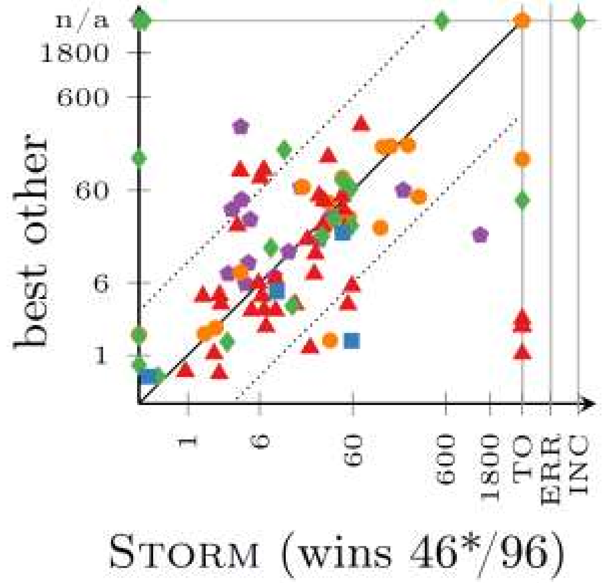


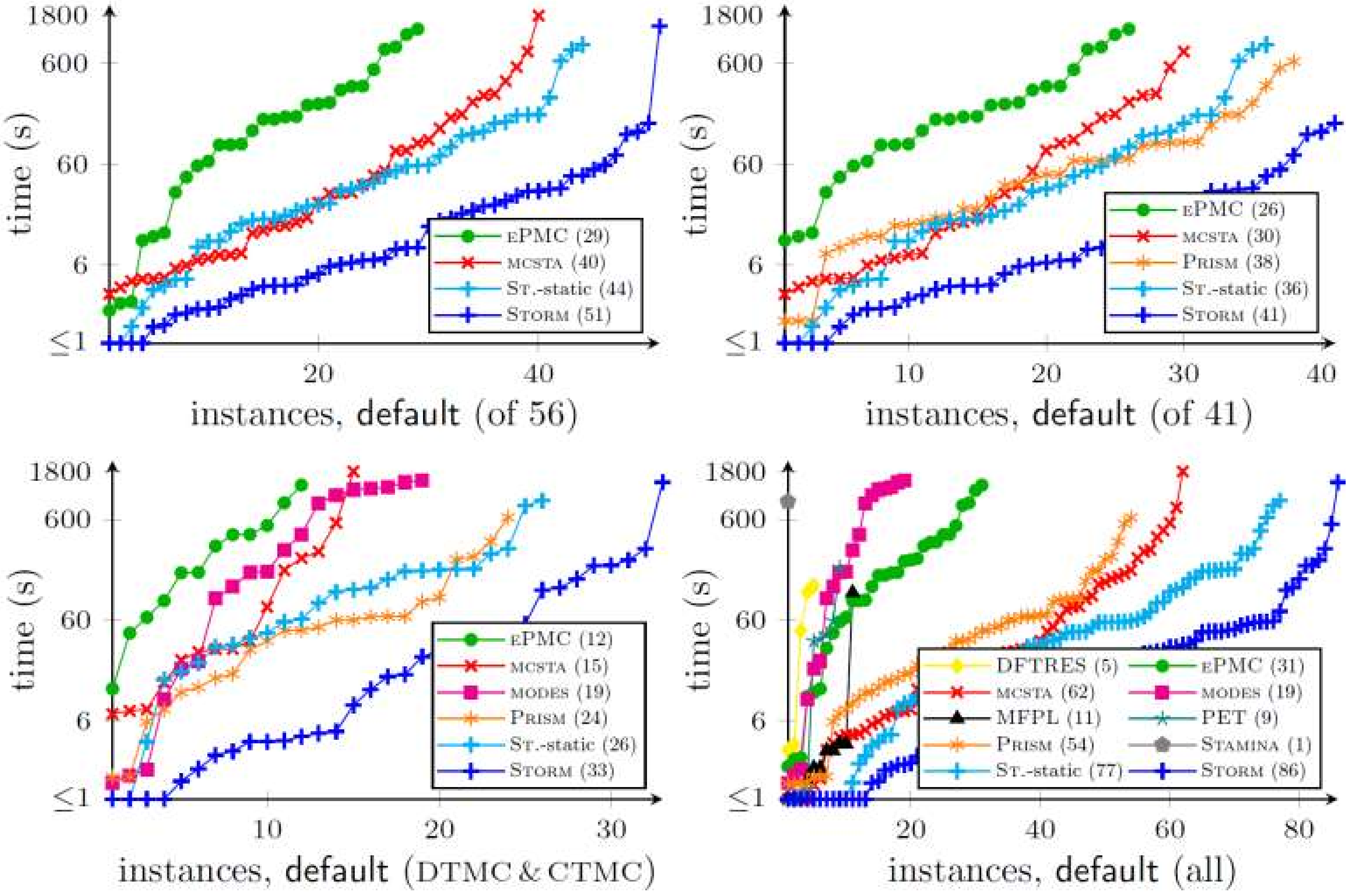
→ quantile plots show whatever you want

QComp 2020: Results

Often ϵ-correct track:
– compare with 2019
– 10′ version useless

Who is the *winner*?





Summary

Quantitative verification: PMC, SMC, and hybrid approaches

Challenges: *algorithm bugs* *unsound algorithms* **specific to quantitative setting**

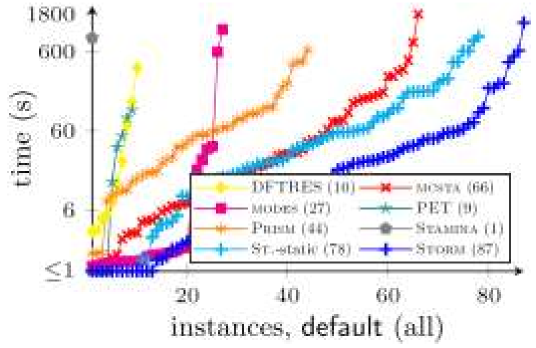
implementation bugs *statistical error*

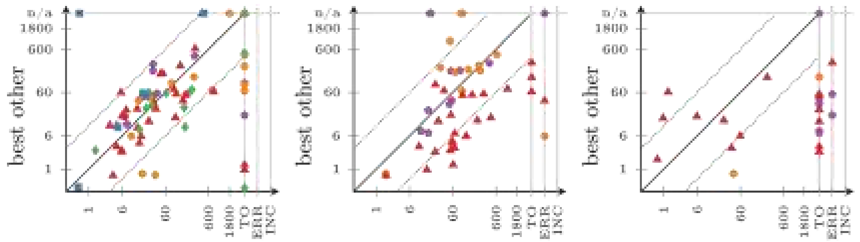
no exact results $|v - \bar{v}|/v \leq \epsilon$ $\mathbb{P}(|v - \hat{v}| > \epsilon) < 1 - \delta$ *floating-point errors*

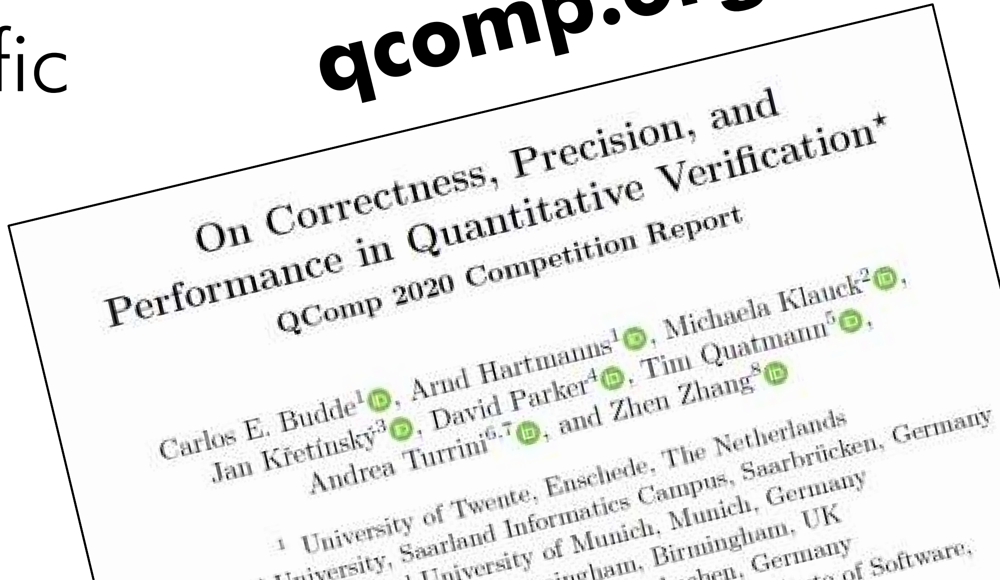
QComp 2020: 5 tracks 100 benchmarks

9 tools default + specific

qcomp.org






On Correctness, Precision, and Performance in Quantitative Verification*

QComp 2020 Competition Report

Carlos E. Budde[1], Arnd Hartmanns[1], Michaela Klauck[2], Jan Křetínský[3], David Parker[4], Tim Quatmann[5], Andrea Turrini[6,7], and Zhen Zhang[8]

[1] University of Twente, Enschede, The Netherlands

+ a tuned STORM

FormaliSE 2022

10th Int. Conference on Formal Methods in Software Engineering

Co-located with ICSE 2022
May 22-23, Pittsburgh, USA

Deadlines (tentative):
- Jan 20: paper submission
- Jan 27: artifacts (voluntary)

Papers: 10 pages, ACM format

…*more info at* ***formalise.org***

Advertisement

RRRR 2022

1st Workshop on Reproducibility & Replication of Research Results

Co-located with ETAPS 2022
April 2, Munich, Germany

Deadlines (tentative):
- Feb 1: short papers (6 pages)
- Feb 15: extended abstracts

Informal proceedings,
extended papers in STTT

…see ***qcomp.org/rrrr/2022***

Summary

Quantitative verification: PMC, SMC, and hybrid approaches

Challenges: *algorithm bugs* *unsound algorithms* **specific to quantitative setting**

implementation bugs *statistical error*

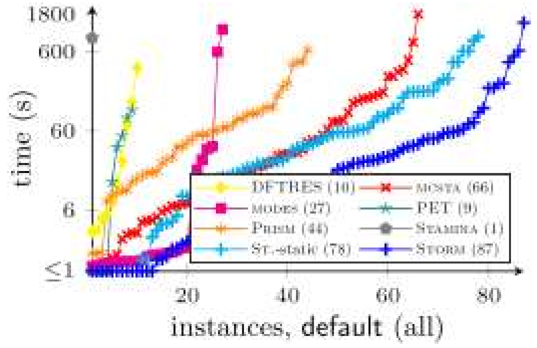
no exact results $|v - \bar{v}|/v \leq \epsilon$ *floating-point errors*

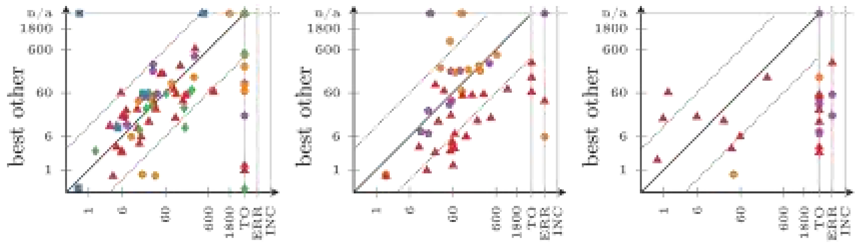
$\mathbb{P}(|v - \hat{v}| > \epsilon) < 1 - \delta$

QComp 2020: 5 tracks 100 benchmarks

9 tools default + specific

qcomp.org





+ a tuned STORM


On Correctness, Precision, and Performance in Quantitative Verification*

QComp 2020 Competition R

Carlos E. Budde[1], Arnd Hartmanns[1], Micha

Jan Křetínský[3], David Parker[4], Tim Qu

Andrea Turrini[6,7], and Zhen Zhang

[1] University of Twente, Enschede, The N