Benchmarks and applications

Advancing Verification Competitions as a Scientific Method

Lorentz Center, February 19th, 2019

Outline

Collect: Sources

Sources

- participants inputs: mandatory/limited
- organizing committee: limits?
- front-end applications: limits, scramble
- generation based on problem specification
- publications
  - case studies
  - theoretical results

Collect: Input format

Problem/property input

- fixed: standards?
  - processing: user vs tool
  - stability
  - compactness
- various: pre-processors

Problem/property meta-information

- origin: license?
- expected result

Validate

Method

- typing
- analyzing statically: class/difficulty/size
- analyzing dynamically: cross checking tools
- "instance carrying proof"

Meta-information given and synthesized

- kind/class/etc
- difficulty
- expected result

Classify

Classification Criteria

- by property checked
- by problem's complexity
- by technique required
- per origin (e.g., SMT-COMP family)

Method

- directories
- specifications (e.g., regular expressions)

Select

At running time

- fixed size versus full set
- avoid biases: overfitting, ...
- detect fraud

To store

- current set
- older versions

Communicate

Archiving

- current set
- older versions

Licensing

- of origin
- of competition

Publishing

- external: fact checking
- with competition

Questions

What benchmarks are interesting to collect

- beyond triviality
- realistic i.e. based on credible applications
- forward looking
- backward looking: keep track of history
- configurable to be scalable
- discriminate tools
- useful for community

Questions

What do you want to avoid to collect

- overfitted
- unrelevant

What benchmarks should be selected for competition

- Highlights best
  - tool (engineering and technique)
  - technique
  - combination of techniques
- Satisfies participants and encourage them to continue
- Detects bias and fraud
- Decision made by the jury?