



Mihály Dobos-Kovács <dkmisu@gmail.com>

SV-COMP 2022 notification for paper 16

1 message

SV-COMP 2022 <svcomp2022@easychair.org>
 To: Mihály Dobos-Kovács <dkmisu@gmail.com>

Thu, Jan 27, 2022 at 8:48 AM

Dear Mihály Dobos-Kovács,

It is my pleasure to inform you that the SV-COMP jury has accepted your paper to be published in the TACAS proceedings.

Please carefully address all suggestions by the reviewers.

Take note of the following title requirements:

- Tool name must be explicitly mentioned
- suffix must be "(Competition Contribution)"
- ideally, a title has the form
 <tool name>: <descriptive title> (Competition Contribution)

Below the bib entry for the competition report.

I will cite all your reports in my competition report.

If you want, you could consider to also cite back my report, which contains a description of the competition, all the results, scoring schema, etc. --- Lots of complementing and useful information.

I will send another e-mail regarding submission of the camera-ready papers.

The page limit is 4 pages *without references*.

It is a good idea to add references to related tools
 (I will later send a list of bib entries for all competition contributions).

Please add the DOIs for all papers that you cite and that have already a DOI.

Best regards,
 Dirk

```
@InProceedings{SVCOMP22,
  author = {D.-Beyer},
  title = {Progress on Software Verification: {SV-COMP 2022}},
  booktitle = {Proc.\ TACAS},
  year = {2022},
  publisher = {Springer},
}
```

SUBMISSION: 16

TITLE: Theta: portfolio of CEGAR-based analyses with dynamic algorithm selection

----- REVIEW 1 -----

SUBMISSION: 16

TITLE: Theta: portfolio of CEGAR-based analyses with dynamic algorithm selection

AUTHORS: Zsófia Ádám, Levente Bajczi, Mihály Dobos-Kovács, Ákos Hajdu and Vince Molnár

----- Overall evaluation -----

SCORE: 1 (weak accept)

----- TEXT:

Paper Review:

Theta is a tool which uses strategy selection for program verification. It selects a strategy using the type of the arithmetic operands in the program

(first strategy for floats, second for bitvectors, and third for integers). The last strategy for integers further checks whether the program contains loops, its cyclomatic complexity, number of inputs read from the environment, and the number of program variables to select appropriate verification configurations. The strategies have been determined using domain knowledge.

The tool Theta this year has implemented several changes as compared to last year. Last year it implemented only one strategy. This year the portfolio has been expanded to accommodate a few more abstraction and refinement techniques, support for verification of concurrent programs, an interface to allow several SMT solvers (last year only Z3 was supported), dynamic checks to avoid infinite refinement loops, and LLVM has been replaced by the ANTLR parser to enable better abstractions.

The paper describes the results, strengths and weaknesses of Theta at length but touches certain points without going into much details. Now there are new page limits, and about half more page available to elaborate. For example, more details on the handling of concurrent programs can be provided (I could not see any reference which presents these details).

A few more suggestions:

1. On SV-COMP 2022 --> In SV-COMP 2022
2. reasoning on the source-level --> reasoning at the source-level
3. "support for SMT-LIB-compliant solvers along with interpolation"
--> this sentence is not clear to me. I understand the first part. As for the second part, wasn't interpolation supported last year also?
4. direct translation --> by direct translation do you mean a translation not in the SSA form?
5. 6166 --> 6163.
6. first configuration --> by first configuration did you mean Mf/E/N?
7. inner timeouts --> do you mean 'internal timeouts'?

Comments are as follows:

1. Fulfills the license requirements -- Yes -- Theta uses apache license for its own libraries which satisfies the competition requirements. The developers have presented the licenses of libraries in LIB_LICENSES.md. All used libraries satisfy the license requirements. As do the licenses of the solvers boolector, cvc4, mathsat and z3 used within Theta.
2. Contributors -- Yes -- The names of all contributors appear in the github repository. The tool also has a separate file 'CONTRIBUTORS.md' that contains the names of all its contributors. But Zsófia Adam's name is missing. It should be added there.
3. Significant conceptual or technological difference with other candidates -- Yes -- Given the submitted abstract, it looks like the only tool performing dynamic algorithm selection in sv-comp'22 while the other tools perform strategy selection.
4. Benchmark definition -- No extra parameters besides bit architecture -- Yes
5. Verifier archive -- contains license -- Y
-- contains readme -- Y

In my opinion, Theta qualifies for sv-comp 2022.

----- REVIEW 2 -----

SUBMISSION: 16

TITLE: Theta: portfolio of CEGAR-based analyses with dynamic algorithm selection

AUTHORS: Zsófia Ádám, Levente Bajczi, Mihály Dobos-Kovács, Ákos Hajdu and Vince Molnár

----- Overall evaluation -----

SCORE: 2 (accept)

----- TEXT:

Summary:

Theta is a tool for CEGAR-based explicit value and predicate analysis. The main advantage of Theta is its ability to iterate through many tool configurations (sequential portfolio) dependent on the given verification task (algorithm selection).

In addition, Theta adapts its verification strategy based on dynamic information gained throughout the verification run.

Review:

The paper is well-written and presents some interesting insights/experiences for the verification community. Therefore, I vote for accepting the paper.

Some remarks:

- Figure 2 is difficult to follow without a legend. It could be beneficial to highlight the type of analyses (e.g. value analyses or predicate analysis) via color or form.

- I am not sure whether the name "dynamic algorithm selection" is really fitting here. As far as I can tell, algorithms are selected based on static program characteristics. I would expect that a dynamic algorithm selector would adapt to current state of the verification run. It is not obvious how this happens inside Theta.

Tool review -----

Fulfills the license requirements:

Yes

Person is a contributing designer/developer of the submitted competition candidate or got approval from organizer to contribute the tool:

In general, it seems that the tool authors are submitting their tools.

However, it is confusing that not all authors that are registered here are also contributors.

Significant conceptual or technological difference with other candidates:

Combines CEGAR based verification with dynamic algorithm selection

No extra parameters besides bit architecture:

Yes

Contains license:

Yes

Contains readme:

Yes. It would be nice if the authors also include some instructions how the tool can be executed in the README.

Allows to have verifier in repository / publicly available for download:

Yes

Allows replication and evaluation by anybody (not only the SV-COMP organizer):

Yes

Does not place any restriction on the verifier output (log files, witnesses):

Yes

Allows any kind of (re-) distribution of the unmodified verifier archive (especially via GitLab):

Yes

----- REVIEW 3 -----

SUBMISSION: 16

TITLE: Theta: portfolio of CEGAR-based analyses with dynamic algorithm selection

AUTHORS: Zsófia Ádám, Levente Bajczi, Mihály Dobos-Kovács, Ákos Hajdu and Vince Molnár

----- Overall evaluation -----

SCORE: 2 (accept)

----- TEXT:

Paper review

The authors present Theta, a CFA/CEGAR-based tool integrating a portfolio of various analysis techniques (e.g., abstract-interpretation with various domains, interpolation, predicate abstraction). Their current version also supports a custom configuration selection algorithm that, depending on properties of the program, selects different configurations with different time limits sequentially.

The paper is well-written and well-structured and gives a good overview over the new configuration selection algorithm.

There are perhaps two things that could be improved:

- * In Sec. 2 you state that CVC4 and Mathsat were essential to tackle bitvectors and floats, but you do not quantify this success. Can you say how many examples could only be solved by using these solvers?.
- * You say that you added support for (sequential) concurrency and target ConcurrencySafety, yet you do not say much about the way you do that. Can you elaborate on that? You still have one page.

Overall, I like the paper and recommend acceptance.