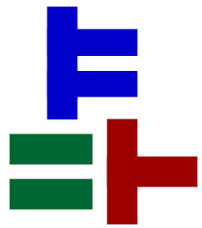
COST action EuroProofNet. WG3 Program Verification

Mădălina Erașcu

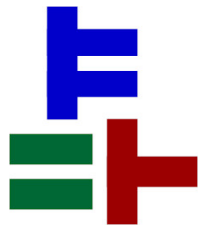
September 18th, 2025



EuroProofNet[1]

Boost the interoperability and usability of proof systems

- Express new proof systems in the Dedukti logical framework
- Promote the output of checkable proofs from automated provers
- Make program verification more effective and accessible
- Gather proofs translated in Dedukti into a database
- Provide tools for searching large libraries of formal proofs
- Develop the use of machine learning techniques in proofs
- Develop a modular theory of type theories
- Develop natural or controlled languages in proof systems

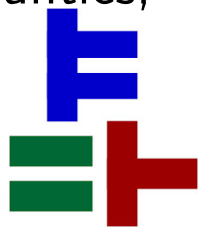


[1] Slide courtesy of Frédéric Blanqui

WG3 - Program Verification

Goal:

- Make program verification more effective and accessible.
- WG3 - beyond the state-of-the-art
  - Verification approaches can envisage new applications and integration of proof systems to overcome challenging problems that combine features that are better expressed in different logics.
  - Scalability and usability of verification techniques can be improved thanks to the exploitation of synergies among different verification tools.
  - Make verification techniques more successful by taking advantage of advances on interoperability between automated and interactive theorem proving, the mathematical formalisation of program semantics, and type theory.

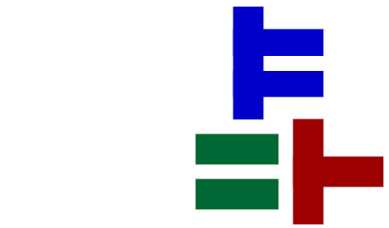


Back in 2023: WG3 Timisoara meeting

- Bring together members of the different communities.
- Create an excellent and inclusive network of researchers in Europe.
- **Discussions** to identify steps towards achieving the action's goals in particular **Deliverables D5 and D6**.
- **Talks** in order to **transfer knowledge** in terms of expertise.

Deliverables:

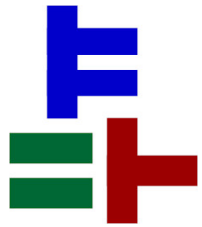
- **D5 (month 18)**: Comparison of the approaches used in the Software Verification competition SV-COMP.
- **D6 (month 24)**: Software prototype for the inference of program specifications as logical axioms.



Output of D5

In the Timisoara meeting, we decided:

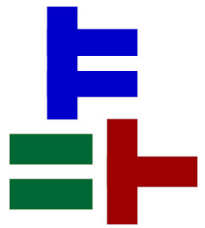
- The deliverable will be a wiki page
- Tool inventory will try to follow the following format[2]
  - What inputs are supported?
  - What properties can be verified?
  - What are the tool?s main techniques for the supported (input, property) pairs?
  - What external tools are used? (compilers, SMT solvers, etc.)
  - What is the tool's URL?
  - What is the ?canonical reference? to a system description?
- We will not be restraint to tools from SV-COMP but extend to tools in the Termination Competition and tools analyzing the termination and complexity bounds.



[2] Thanks to Carsten Fuhs

Output of D5 (cont'd)

Competition on Software Verification (SV-COMP)[3] is an anual (since 2012) competition held in conjuction to ETAPS.



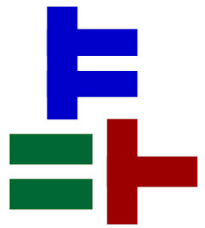
[3] https://sv-comp.sosy-lab.org/

[4] https://termination-portal.org/wiki/Termination_Competition

Output of D5 (cont'd)

Competition on Software Verification (SV-COMP)[3] is an anual (since 2012) competition held in conjuction to ETAPS.

Termination Competition[4] is an anual (since 2004) competition held in conjuction to conferences/workshops on automated reasoning.



[3] https://sv-comp.sosy-lab.org/

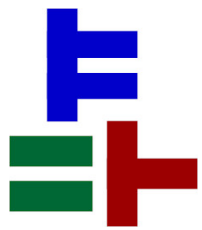
[4] https://termination-portal.org/wiki/Termination_Competition

Output of D5 (cont'd)

Competition on Software Verification (SV-COMP)[3] is an anual (since 2012) competition held in conjuction to ETAPS.

Termination Competition[4] is an anual (since 2004) competition held in conjuction to conferences/workshops on automated reasoning.

- The competition introduces a shared benchmark suite that enables objective evaluation of different software verification tools.
- It highlights and rewards the work of developers, especially students and researchers, who contribute technical innovations.
- By establishing public benchmarks and showcasing state-of-the-art methods, it supports progress and collaboration in software verification research.



[3] https://sv-comp.sosy-lab.org/

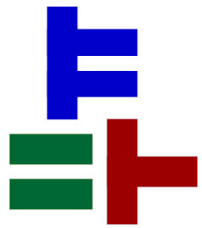
[4] https://termination-portal.org/wiki/Termination_Competition

Output of D5 (cont'd)

Webpage: `https://github.com/EuroProofNet/ProgramVerification/wiki/List-of-tools`

How?

- We asked the participants of the Timisoara meeting and the WG3 members to contribute.

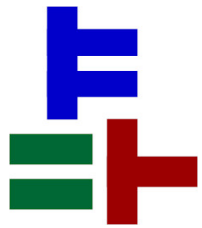


Output of D5 (cont'd)

Webpage: `https://github.com/EuroProofNet/ProgramVerification/wiki/List-of-tools`

How?

- We asked the participants of the Timisoara meeting and the WG3 members to contribute.
- Ask them to contact directly possible contributors.

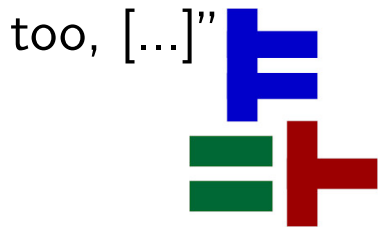


Output of D5 (cont'd)

Webpage: `https://github.com/EuroProofNet/ProgramVerification/wiki/List-of-tools`

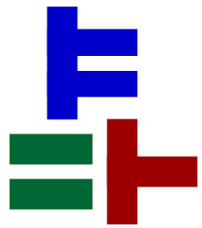
How?

- We asked the participants of the Timisoara meeting and the WG3 members to contribute.
- Ask them to contact directly possible contributors.
- Took one of the papers on the progress of SV-COMP and write individual emails to the authors of the tools found in this paper.
  - Most of them did not answer.
  - "The sv-comp tools are already on Zenodo as well as their individual GitHub sites. Instead of cloning these on your repository too, [...]"



Lessons learned...

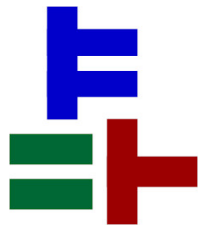
... which might be exploited in the future.



Lessons learned...

... which might be exploited in the future.

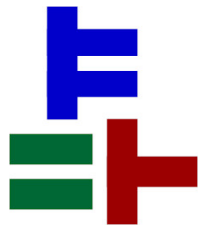
1. Challenge: Researchers have little time, so unless the benefit of having such an intiative is clear, the task can be ignored. Solution State concrete gains (joint paper, citations, inclusion in comparison tables, colaborations, visibility in the community, etc.)



Lessons learned...

... which might be exploited in the future.

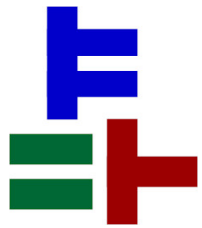
1. Challenge: Researchers have little time, so unless the benefit of having such an intiative is clear, the task can be ignored. Solution State concrete gains (joint paper, citations, inclusion in comparison tables, colaborations, visibility in the community, etc.)
2. Challenge: Ongoing maintenance feels burdensome. Solution: Auto-sync key fields from GitHub/Zenodo.



Lessons learned...

... which might be exploited in the future.

1. Challenge: Researchers have little time, so unless the benefit of having such an intiative is clear, the task can be ignored. Solution State concrete gains (joint paper, citations, inclusion in comparison tables, colaborations, visibility in the community, etc.)
2. Challenge: Ongoing maintenance feels burdensome. Solution: Auto-sync key fields from GitHub/Zenodo.
3. Challenge: New users may find the wiki overwhelming or hard to navigate if the list gets long. Solution: Provide simple navigation aids (categories - by what?, search, tags, or a "start here" guide highlighting the most widely used tools).

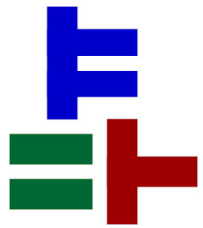


Lessons learned...

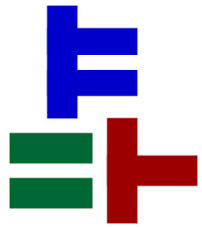
... which might be exploited in the future.

1. Challenge: Researchers have little time, so unless the benefit of having such an intiative is clear, the task can be ignored. Solution State concrete gains (joint paper, citations, inclusion in comparison tables, colaborations, visibility in the community, etc.)
2. Challenge: Ongoing maintenance feels burdensome. Solution: Auto-sync key fields from GitHub/Zenodo.
3. Challenge: New users may find the wiki overwhelming or hard to navigate if the list gets long. Solution: Provide simple navigation aids (categories - by what?, search, tags, or a "start here" guide highlighting the most widely used tools).
4. Challenge: Discoverability is limited if the wiki lives in isolation. Solution: Cross-link with related initiatives (benchmarks, workshops, mailing lists), and add tags/categories so tools can be filtered and searched more easily.

How did you proceed to achieve greater involvement?



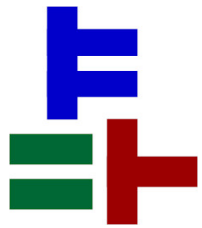
Simmilar initiatives[5]



[5] Thanks to ChatGPT5

Simmilar initiatives[5]

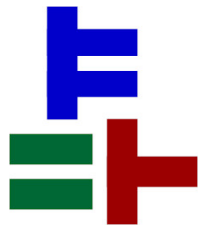
- Formal Methods Tools (FMTools.fyi) `https://www.fmtools.fyi/`



[5] Thanks to ChatGPT5

Simmilar initiatives[5]

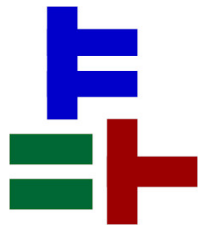
- Formal Methods Tools (FMTools.fyi) `https://www.fmtools.fyi/`
- Awesome Formal Verification `https://github.com/ElNiak/awesome-formal-verification`



[5] Thanks to ChatGPT5

Simmilar initiatives[5]

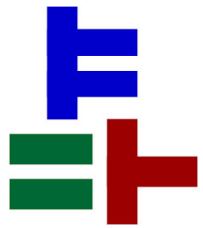
- Formal Methods Tools (FMTools.fyi) `https://www.fmtools.fyi/`
- Awesome Formal Verification `https://github.com/ElNiak/awesome-formal-verification`
- Tool Support for Formal Methods `https://www.fmeurope.org/industry/tool-support/`



[5] Thanks to ChatGPT5

Simmilar initiatives[5]

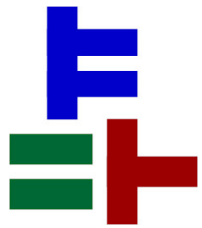
- Formal Methods Tools (FMTools.fyi) `https://www.fmtools.fyi/`
- Awesome Formal Verification
  `https://github.com/ElNiak/awesome-formal-verification`
- Tool Support for Formal Methods
  `https://www.fmeurope.org/industry/tool-support/`
- FM-Tools (by SoSy-Lab)
  `https://gitlab.com/sosy-lab/benchmarking/fm-tools`



[5] Thanks to ChatGPT5

Simmilar initiatives[5]

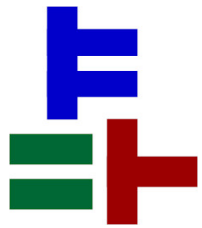
- Formal Methods Tools (FMTools.fyi) `https://www.fmtools.fyi/`
- Awesome Formal Verification `https://github.com/ElNiak/awesome-formal-verification`
- Tool Support for Formal Methods `https://www.fmeurope.org/industry/tool-support/`
- FM-Tools (by SoSy-Lab) `https://gitlab.com/sosy-lab/benchmarking/fm-tools`
- List of model checking tools (Wikipedia) `https://en.wikipedia.org/wiki/List_of_model_checking_tools`



[5] Thanks to ChatGPT5

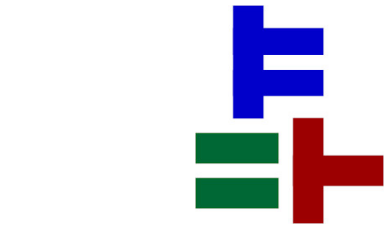
Simmilar initiatives[5]

- Formal Methods Tools (FMTools.fyi) `https://www.fmtools.fyi/`
- Awesome Formal Verification `https://github.com/ElNiak/awesome-formal-verification`
- Tool Support for Formal Methods `https://www.fmeurope.org/industry/tool-support/`
- FM-Tools (by SoSy-Lab) `https://gitlab.com/sosy-lab/benchmarking/fm-tools`
- List of model checking tools (Wikipedia) `https://en.wikipedia.org/wiki/List_of_model_checking_tools`
- Open Hardware Verification `https://github.com/ben-marshall/awesome-open-hardware-verification`



[5] Thanks to ChatGPT5

Program Verification Tools - Community Input Survey



Questions?