

EVALUATION OF RESEARCH

Why provide an evaluation of proposed concepts?

- For readers: To convince
- For us: To find weaknesses in the approach

How to evaluate?

- Competitions ✓
- Experiments ✓
- Data Analysis ✓
- Surveys ✓
- Proofs

10.1 Competitions

- Comparative evaluation
- Participant: Team + Tool
- Team selects best possible tool configuration for task
- Open call for participation + call for benchmark instances
- Ideally: objective ranking (example: SV-COMP; counterexample: VerifyThis)
- Biases: Scoring scheme and benchmark (community agreement)

Example competition types:

- Performance + Precision
- Usability / Expressiveness
- Research projects (e.g., student research competition)

! - benchmark selection
! - no repetition
↳ training mit je 1000
→ non-deterministic still an issue

10.2 Experiments

1. Hypothesis that should be falsified/supported
2. Experiment setup and plan
3. Execution
4. Results
 - Interpretation
 - Effect on hypothesis

[Popper, 1935]

Every hypothesis should be formulated such that it is falsifiable.
→ Causality vs. correlation

Experiments must be available and reproducible.

- Reliability
- Validity
- Objectiveness

Further information (in German): A. Butz and Antonio Krüger: Section 13: Evaluation. In Mensch-Maschine-Interaktion, De Gruyter Oldenburg 2017.

10.3 Data Analysis

- Data already available

http://81.169.143.104/archiv/Verschiedenes/Popper_Karl

1. Data scraping / mining
2. Analysis

Common tools for analysis:

- Jupyter Notebooks ←
- R ←
- Python ←
- Jasp
- gnuplot
- PGF plots (tikz)
- Plotting in python with seaborn

10.4 Surveys

- Interviews
- Questionnaires
- Monitoring/Supervision
- Focus groups

very inductive 2 kinds $\left\{ \begin{array}{l} \text{structured (same questions to all)} \\ \text{open (not determined upfront)} \end{array} \right.$
not inductive purpose, but there could be conditional

10.5 Proofs

- Theorem, Proposition, Lemma
- Proof, proof draft
 - proof by induction
 - proof by deduction
 - proof by construction
 - proof by contradiction
- Statements that are not provable are ^{yet} ‘conjectures’

10.6 Excourse: Numbers in LaTeX

For typesetting numbers in LaTeX (e.g., experimental data), `\usepackage{siunitx}`.

```
\SI{15}{\giga\byte}
\SI{15}{\minute}
```

M
m
↘ meter
min

15GB
15 GB
↑
↘,

15 GB memory
quants unit measure

TECHNICAL HINTS FOR WRITING PAPERS

11.1 Paper Structure

- Abstract
- Introduction + Related Work
- Concepts
- Evaluation
- Conclusion
- Statements (Data Availability, Funding, Acknowledgements)
- References (Dols are must)

11.2 Paper Content

- See review criteria
- Tell the whole story in the introduction, rest is detail

11.3 Methodology for Paper Writing

P.R. Halmos:

There is no recipe and what it is.

The basic problem in writing mathematics is the same as in writing biology, writing a novel, or writing directions for assembling a harpsichord: the problem is to communicate an idea.

To do so, and to do it clearly, you must...

- have something to say,
- have someone to say it to,
- organize what you want to say,
- arrange it in the order you it said in,
- write it, rewrite it, and re-rewrite it several times,
- be willing to think hard about and

φ gives us

- work hard on mechanical details such as diction, notation, and punctuation.

That's all there is to it.

11.3.1 Psychological tricks:

- empty-page effect
- divide and conquer , *stepwise refinement* , *refactoring*
- small steps, incremental , *commit after*
- start, even if you think the outcome will be marginal

11.3.2 Approaches to writing:

- Treat writing a paper like programming/software development
- Source code is precious, should be of high quality and maintainable
- Consider teamwork / shared code

Consequence: Source code of paper should be:

- well readable: structured, organized
- minimal: refactored, avoid repetitions with macros, remove unnecessary comments

Examples:

- Do not write 'CPAchecker', but use `\newcommand{\cpachecker}{CPAchecker}`. If you change typesetting to `\textsc{CPAchecker}`, only single location must be changed
- Do not print numbers of experimental data in table and repeat them in text, but use `\newcommand{\correctAlarmsCpachecker}{238}`. Avoids typos and inconsistencies (also makes automation easier).

Source: Our wiki ←

More sources:

- Armando Fox: Armando's Paper Writing and Presentations Page
- • Donald E. Knuth, Tracy Larrabee and Paul M. Roberts: Mathematical Writing
- Joel E. Cohen: To A Young Scientist
- Mary-Claire van Leunen and Richard Lipton: How to Have Your Abstract Rejected
- Mark Wegman: What it's like to be a POPL referee; or how to write an extended abstract so that it is more likely to be accepted
- P. R. Halmos: How to Write Mathematics
- Simon Peyton Jones: How to write a good research paper, give a good research talk, and write a good grant proposal
- William Pugh: Advice to Authors of Extended Abstracts

11.4 Text editing

- (at least) every sentence on new line
- avoid changing line breaks (do not auto-wrap lines)
- use short lines
- avoid tabs, use spaces
- commit everything to repository that is required to build the project
- also commit the bibliography file (main.bbl)!
- always check diff before push

Example diff to show how short lines are helpful: What is the actual change?

```
- To increase the reliability of these results, validators like \textsc{CPachecker} \cite
↪{CPACHECKER}, \textsc{UAutomizer} \cite{UAUTOMIZER2013}, and \textsc{MetaVal} \cite
↪{MetaVal} use witnesses to reverify programs wtih desirably less effort by trying to
↪reduce the state space by strengthening the used analysis with
+ To increase the reliability of these results, validators like \textsc{CPachecker} \cite
↪{CPACHECKER}, \textsc{UAutomizer} \cite{UAUTOMIZER2013}, and \textsc{MetaVal} \cite
↪{MetaVal} use witnesses to reverify programs with desirably less effort by trying to
↪reduce the state space by strengthening the used analysis with
```

vs.

```
To increase the reliability of these results,
validators like \cpachecker~\cite{CPACHECKER},
\uautomizer~\cite{UAUTOMIZER2013}, and \metaval~\cite{MetaVal}
use witnesses to reverify programs
- wtih desirably less effort
+ with desirably less effort
by trying
to reduce the state space by strengthening the used analysis
with
```

11.5 Comments

- % Text that may be used in the future
- %%% db: Explanation by Dirk Beyer for co-authors
- Each comment starts a new line

11.6 Macros

- `\newcommand{...} [#args] {...}`
- Never use `\def!` Silently overwrites existing macros

11.7 Writing Style

- avoid ambiguity with symbols: ‘a set S of states’, **not** ‘a set of states S’. With the latter, S could also be a state.
- ‘iff’ in conditions (theorems), ‘if’ in definitions

11.7.1 Spaces and Periods

- french spacing: ‘D. Beyer’ -> D. Beyer (single space, not larger) *D. ~ Beyer*
- paragraphs: `\noindent`, `\smallskip`, `\medskip`, `\bigskip`
- hair space between quantity and unit: `33\,GB` (best solved with `siunitx`) *1.5'*

Normal space: 30 GB

Hair space: 30GB

- use short symbols: ‘the state~s’, ‘for all~i’
- citations:
 - Reference is **not** part of sentence, never used as object in sentence. Removing the reference may not destroy the sentence. Bad: “[3] says ...”, “as described in [4]” Good: “as described by Smith [4].”, “as done in HCI [5]”
 - Always use non-breaking space `~\cite{...}` *they described a good technique [4].*
 - Avoid “et al.”

11.8 Types of Hyphens and Dashes

- Hyphen: Conjuncts words *→ model-checking tools*
- En-dash: Minus and range
- Em-dash: Separation of thoughts, insertion

Hyphen: Long-lasting

En-dash: 1–5

Em-dash: runs—if found valid—for a long time

11.9 BibTeX

- Use proper names for entry keys
 - R-textbook
 - Fool-Name, Author-Name + Year
 - Conference
 - Best: Memorizable abbreviation
- Entry data should be in consistent style and formatting
- Use ‘annot’ field for short explanation why paper is relevant

url_ = ...
doi = { } ;
doinone = {last checked 2022-07-19}

11.10 Repository Organization

- Start repository as soon as serious work on topic exists (e.g., whiteboard screenshot)
- Commit each document
- Main source file: main.tex
- Split large documents into sections. For example: main.tex, abstract.tex, intro.tex, ...
- Sub directories:
 - archive/
 - * 2021-03-02_CAV07-submission.pdf
 - * 2021-07-05_CAV07-published.pdf
 - * 2021-07-05_CAV07-Copyright-Form.pdf
 - * 2021-05-05_CAV07-rebuttal-answer.txt
 - paper/
 - * main.tex
 - * ...
 - talk/
 - * main.tex
 - * ...

CAV 21 - submit
CAV 21 - final
JAR 23 - submitted

GRANT PROPOSALS

Proposals for Financial Support

- Is it a good idea to support the project?
 - importance
 - potential impact
 - How does it change the world?
- Is the PI qualified to conduct the project?
 - previous research in the area
 - community service
- Does the PI know how to do it?
 - work packages
 - time schedule
- Does the PI have the resources to do it?
 - Are there people available for the project?
 - Is the assignment of work packages to people matching?