Exercises in Academic Writing

Clara Löh

Abstract. Research papers are a fundamental means to share and archive research results within the scientific community. The aim of this nano-course is to learn how to write a (good) mathematical paper. In particular, we will highlight some peculiarities that are specific for mathematical writing. As writing is not a spectator sport, these notes also contain a series of exercises.

Contents

May 15, 2017. This work was supported by the GRK 1692 *Curvature, Cycles and Cohomology* (funded by the DFG, Universität Regensburg).

Introduction

Research papers are a fundamental means to share and archive research results within the scientific community. Moreover, journal publications seem to be the most valuable currency in the market of CV's of junior researchers. Therefore, writing good research papers will both benefit the community and the author.

The aim of this nano-course is to learn how to write a (good) mathematical paper. The main ingredient of a good research paper is good research. This will not be the topic of this course. Instead, we will assume that we have obtained an interesting result and we will focus on the craft of writing up our discoveries and ideas.

Writing a research paper has several aspects:

- *The global level.* Research papers are composed out of standard building blocks such as an abstract, an introduction, references, ... This global structure needs careful organisation and consideration.

- *The local level.* Also on the small scale, mathematical research papers follow certain rules concerning language and style.

- *The peer-review process.* Publishing a paper in a journal starts with the selection of an appropiate journal (i.e., in terms of scope, so-called reputation, copyright policy, . . .). The actual peer-review process involves interaction with the editors, waiting for the referee report, and potentially interaction with the referees (usually through the editors).

Your supervisor(s) will guide you through the peer-review process (which is rather sensitive to the specific research field). If they forget to do so, you should explicitly ask them or a senior postdoc in your group for this kind of advice. The choice of journal and audience will affect the way the paper should be written, but we will assume that this choice has already been made.

In this course, we will focus on the global and local structure. In particular, we will highlight some peculiarities that are specific to mathematical writing. As writing is not a spectator sport, these notes also contain a series of exercises.

Disclaimer. Almost everything that will be discussed in this course might seem obvious (or obviously nonsense, depending on the reader's preferences) in one way or the other; the main point is to think about all of these issues *at all.* Many aspects of writing are matters of taste and style and as such are highly subjective. This course is biased by my personal preferences.

1 An axiom for academic writing in mathematics

In this course, we will adopt the following axiom:

Axiom 1.1. Mathematical papers are software.

Let me explain why this axiom is not completely nuts, but a reasonable point of view that will lead to improved writing:

- Software in the *literal sense*: Mathematical papers currently tend to be written in LaTeX, which is a programming language.
- Software in the *structural sense*: Mathematical papers contain code in natural language (namely definitions, theorems, and proofs) that is parsed, interpreted, executed, and reused/linked by humans.

In fact, this point of view of mathematical proofs as programs is formalised in the Curry-Howard correspondence [22] and is used in proof assistants such as Coq [3].

Following this line of thought, we can immediately derive consequences for the infrastructure and the contents of mathematical papers.

1.1 Consequences for the infrastructure

Mathematical papers usually are created as LaTeX source code. Hence, we should treat this LaTeX source code as source code. In particular, we should not forget to steal tools and best practices from software technology.

- *Save early, save often!* Writing source code requires tools for backup, logical version control, and cooperation with coauthors. Fortunately, an "eierlegende Wollmichsau" is available: version control systems (e.g., git [7], hg [9], darcs [4], ...).
- *Build professionally.* Once a LaTeX document gets complex (e.g., large bibliography, index, etc.), it is convenient to automate the build process to facilitate creation of the pdf document. At this point, tools such as make [17] or latexmk [16] are convenient.
- *Share!* Finally, the finished manuscript wants to be read and hence needs to be shared with the scientific community. In order to reduce the delay through sluggish journals and in order to maximise the group of readers, it is highly recommended to first publish the paper as a preprint on an appropriate preprint server such as the arXiv [1].

1.2 Consequences for the contents

Viewing mathematical papers as programs in the structural sense uncovers the following issues with mathematical writing:

As the “programming language” is a mixture of natural language and the formal language of logic and set theory. So, the language is only partially specified.

Moreover, also the execution of “mathematical code”, i.e., the interpretation, is only partially specified.

As a consequence, verification of correctness of most mathematical papers can only be done by hand (even though, theoretically, this could be automated) and thus error-prone. Therefore, it is imperative to strive for clarity and transparency in mathematical writing and to try to find the right balance between formality and readability.

A central virtue of mathematical research is that results by other researchers can be reused and built upon. In the context of programming, this corresponds to a module system and linking of librariers. In our context, this is modelled by citations; while common practices and traditions have developed, these are subject to change. In contrast to programming, in research only very few works exist that play the role of “pure” libraries (with the notable exception of the Bourbaki series).

While modern programming frameworks provide powerful tools to search for existing functionality, searching for specific results in mathematics easily turns out to be a difficult task. Basic tools are mathscinet [19], google [8], and mathoverflow [18], but every mathematician knows at least one story of accidental rediscovery, because the original result has been overlooked. While Wikipedia [28] can be useful to some extent, it certainly does not qualify as a scientific source and does contain articles with serious mathematical mistakes.

A point related to this problem is that there is no systematic checking for deprecated/wrong results. Thus, state of the art research in mathematics requires to stay active and well integrated into the community.

In the following, we will discuss some simple techniques that help to achieve lucid writing: on the global level (Section 2), on the local level (Section 3), and the basic building blocks (Section 4).

2 Global level

Mathematical research papers are not isolated entities, but will only come truly alive once they are cited and reused in future research. Therefore, we should keep an eye on providing a well-designed interface to our fellow researchers.

The programming analogue is that programs need to declare a proper interface with carefully chosen public functions and functionality.

In the same way, a research paper needs to prominently declare the main results (with self-contained formulations) as well as carefully selected intermediate steps and lemmas.

One principle in the division of larger units into smaller units is the proper use of abstraction. In the same way as programming code should never be duplicated also the duplication ("copy-paste") of arguments should be avoided by all means. Instead, it is much better to search for a common abstraction that allows to derive all necessary cases by specialisation.

By the same principle, the following formulations should be avoided: "the same argument as in the proof of ... shows" or even worse "similarly to the proof of ... in ... one can prove that ...". While this might be convenient for your current paper, think of the poor researcher that wants to generalise/modify your arguments to another setting.

One expository problem that does not occur in programming is that mathematical papers are linear texts. The partial order given by the intermediate steps and their logical dependencies is just a directed graph (without directed cycles!), but not necessarily a disjoint union of directed linear graphs. Hence, flattening the logical structure of the arguments into a linear text might be a non-trivial task.

In programming, powerful module or package systems take care of the administration of dependencies between different modules. In scientific publications, dependencies arise through citations. Keeping in mind that cited results may turn out to be wrong, it is particularly important to mark all used results through adequate citations.

The typical building blocks of mathematical papers will be discussed in detail in Section 4.

3 Local level

We will now proceed to the local level.

Identifiers

Every decent compiler will complain if undefined identifiers are used, if identifiers are defined multiple times, or if identifiers are defined but never used. The same applies to identifiers in mathematical papers (in order to minimise the danger of confusion and misunderstandings):

- All identifiers need to be introduced.
- No identifier should be re-defined.

- No unnecessary identifiers should be introduced.
- No notation should be implicit.

Establishing good notation can be a big help for both the author and the reader.

Locality

Locality refers to the property that certain parts of a research paper should be comprehensible without reading all of the paper in linear order.

In particular, theorems should be self-contained; i.e., all hypotheses and assumptions should be contained in the statement of the theorem. This will help future readers that jumped into your paper through a reference from another paper to correctly apply your results.

Sometimes, a theorem needs many lengthy hypotheses that would clutter the statement of the theorem. Or there are multiple theorems that use the same long list of hypotheses. In such cases, it is worth outsourcing the setup into a separate, explicit, setup environment (or into a new definition) that can be referred to.

For auxiliary lemmas and intermediate steps, it might be more convenient to fix some notation/hypotheses at the beginning of the corresponding section. However, in general, it is good to strive for locality and a little bit of redundancy.

Obvious

The word “obvious” and all its relatives and derivatives should be handled with care. It is best to avoid them completely and to replace them with a more descriptive formulation.

Citations

Citations are an integral part of scientific publishing. As mentioned in the discussion of global aspects, citations document the dependencies between different sources. However, they are often treated badly or abused by authors. In order to make your paper accessible, reusable, and transparent, you should stick to the following rules:

- Everything that is not obvious (sic!) requires a proof or a citation.
- Make sure that the cited result can be applied in your situation. In case this way of applying the cited result is not straightforward, an explanation is in order.

- Give a precise citation (including theorem number or page number etc.) that will actually enable the reader to find the result in the cited source.

- In general, original sources should be cited. However, in many cases, these original sources are not the easiest to understand. Therefore, it can be helpful to add a citation of a good textbook.

- The citation labels are not part of the text and hence cannot be used as words. Therefore, sentences such as "In [11], you will find a lot of dirty tricks." should be avoided. Instead, a more descriptive formulation should be used: "Knuth's TEXbook [11] contains a lot of dirty tricks."

Reconstructing references after completing your research can be a daunting and time-consuming task. Therefore, it is helpful to make it a habit to write down useful references whenver you use them in your preliminary notes and fragments.

Macros are your friend

Mathematical papers usually are created as LATEX source code and LATEX is a (Turing complete!) programming language. Therefore, we should make use of this power.

Many non-trivial tasks can be performed by macros. For example, the layout, all sectioning commands, etc. are taken care of by macros.

A particularly important class of macros are environments for theorems, lemmas, proofs, etc.. A good framework for such environments is provided by the amsthm-package.

However, we should also use macros for all other code that is likely to be used more than once (e.g., complicated mathematical terms that appear several times). This will save a lot of time during typing, decrease the number of typos, and enable us to apply global changes easily.

In addition, it is convenient to define macros for comments etc. that help to organise the writing process (but will not appear in the final document).

Symbols

Mathematical symbols naturally play an important role in mathematical writing. However, they should be handled with some care.

- Try to find a good balance between using too few and too many symbols. Spelling out equations in prose is not helpful to the reader; neither is it to obfuscate a simple idea with heavy use of symbols.

- Don't start sentences with symbols (because of readability).

- Symbols that do not belong together need to be separated by at least one word.

- Use symbols only according to their specification. In particular, logical quantifiers (such as $\forall$ and $\exists$) should only be used in logical formulae (and not as abbreviations in a text block). Moreover, as they bind variables, they need to be put in front of the variable's scope!

  In case you are not yet convinced: Nobody would ever dare to write something like
  $$\sum_{j=1}^{n} a_{jk} \prod_{k=1}^{m},$$
  but surpisingly many authors happily write logical gibberish such as
  $$\exists_{j\in\{1,\dots,n\}} \quad A(j,k) \quad \forall_{k\in\{1,\dots,m\}}.$$

- Use the correct LaTeX symbols. For example, angle brackets $\langle$ and $\rangle$ should be typeset via \langle and \rangle (and *not* by the binary operators $<$ and $>$). Another common example is that one should use \llbracket and \rrbracket (from the stmaryrd package) to typeset the double brackets $\llbracket$ and $\rrbracket$ for rings of formal power series.

  A convenient way to search for LaTeX symbols is the detexify tool [5].

Graphics

Many definitions and arguments gain from a graphical representation. The following general guidelines apply:

- Whenever possible, use vector graphics.

- The font (and its size) used in figures should be the same as the font of the main text. This is particularly important for formulae.

- The thickness of lines should be compatible with the font of the main text.

- The arrow tips in figures should match the arrow tips used in formulae.

- Floating figure environments need labels and captions.

Convenient tools to create such graphics are tikz [25] and METAPOST [12].

Jokes

There are memorable jokes in mathematical research papers. If you want to include a joke into your own paper, please make sure that it is actually funny and that it can be understood by the reader.

Language

It goes without saying that a research paper should be grammatically and orthographically correct. There are many tools that help with that – spell-checkers, dictionaries, grammar references, etc. [21, 20].

Moreover, there are many words and phrases that need extra care [13, 6], for example:

- *this.* Using "this" without further attributes is dangerous, because it can lead to ambiguities. It is therefore highly recommended to expand "this" to a more descriptive phrase.
- *important.* The word "important" is too generic to convey useful information. Usually, it is easy to replace it by a more precise formulation. Or to remove it.
- *for any.* The quantifier "for any" can lead to misunderstandings (depending on the context, it can be understood as "for one specific" or "for all"). It should be replaced by a precise quantfier.
- *canonical/natural.* The words "canonical" and "natural" have precise meanings: Something is "canonical" if it does not depend on specific choices or additional data; something is "natural" if it is compatible with morphisms (i.e., a kind of functoriality).

4 Building blocks

We wil now discuss some of the basic building blocks of research papers. In order to understand how to write these building blocks, we should first think about the function of these building blocks and how readers interact with research papers.

Most readers approach research papers in the following order:

- title/author(s) (!)
- abstract

- introduction
- references
- main body of the paper
- preliminaries

At each of these stages, readers might drop out because they are not convinced that the paper is relevant or interesting enough for them. So, every one of these building blocks should encourage the reader to continue reading!

One major exception are readers that follow a reference to a specific theorem in the paper in question – they will usually jump directly to that particular spot in the paper. However, also such readers will benefit from a paper whose standard building blocks are carefully written.

In the following, I will list the standard building blocks in a natural writing order.

4.1 The main theorem

The main theorem should be self-contained. The theorem should be stated in a (locally) optimal way, i.e., there should not be an "obvious" generalisation possible. Moreover, the theorem should be formulated in such a way that others can use it with convenience.

In order to make sure that the main theorem is plausible (independently of the proof) it is worthwhile to test the theorem in different classes of examples.

4.2 The title

The title should concisely summarise the main result or aim of the paper; ideally, without being too technical.

4.3 The abstract

The abstract should give a brief (i.e., one or two paragraphs) overview of the context, main results, main techniques, and applications of the research paper. The abstract should be not too technical and as self-contained as possible. In particular, an abstract should neither contain references nor complicated formulae.

One should always keep in mind that the abstract does have the power to attract people to read your paper. Or to scare them off.

One way of writing abstracts is to keep the following recipe in mind (in analogy with the advice by Swales and Feak on writing introductory sections [24, Unit Eight]):

- *Establishing a research territory.* The first sentence(s) should give some context, relating the paper to previous work and established research directions.
- *Establishing a niche.* The next sentence(s) should point out an open problem or a natural direction of research in this context.
- *Occupying the niche.* Finally, the main findings of the paper should be outlined. This consists of a brief description of the main results, the methods used to prove these results, and whenever possible also applications of these results.

Writing a first draft of the abstract in an early stage has the advantage that one can use the abstract for guidance of the whole writing process.

4.4 Proofs

A proof should be correct and complete, it should be easy to follow (i.e., the arguments should be divided in small, checkable units), and it should be plausible to a human reader.

Therefore, it is helpful to follow some guidelines (both to the top-level proof and to proofs of smaller steps):

- Explain the main idea/strategy of the proof.
- Subdivide the proof into manageable steps, move auxiliary calculations etc. to separate lemmas.
- Are there dependencies between the intermediate steps? What is a good linear exposition of these steps? Is there an opportunity for abstraction in order to avoid repetition?
- Guide the reader through the structure of the proof.
- Try to be as transparent as possible. Obfuscation is great in some coding contests [10], but *not* in mathematical writing. Transparent arguments are less error prone and more user-friendly!
- Would it help to illustrate the terms/arguments by figures?

A structured way of writing proofs has been advertised by Lamport [14, 15].

4.5 Preliminaries

It can be convenient to collect preliminaries and basic notation needed for the main theorem and the main proofs in a separate section of the paper. At this point, one should be concise and focus on the material needed for the paper.

4.6 Bibliography

In addition to references that are logically needed for the proofs in the paper, one also needs to put the work into context. Therefore, the bibliography will also contain relevant related work.

Please make sure that all names and titles in the bibliography are spelled correctly and that the bibliographical information is complete. Works that are "in preparation" should (whenever possible) *not* be put into the bibliography.

4.7 Applications

An application of a theorem is a consequence of the theorem that does not involve the theorem directly in its formulation. Presenting applications will increase the value of your research – the applications prove that your result is useful.

In some cases, it is rather difficult to give applications. In these cases, one should at least give some examples of the theorem. Moreover, these examples also help you to check the correctness of the theorem.

4.8 The introduction

The introduction is the crown jewel of any research paper. Unfortunately, writing this part of the paper is particularly demanding:

Theorem 4.1. *It is impossible to write a perfect introduction.*

Proof. Assume for a contradiction, that it would be possible to write a perfect introduction. Such an introduction would fulfill the following tasks:

- Explaining the context of the result, discussing previous and related work.
- Explaining why the main problem studied in the paper is a reasonable problem.
- Stating the main results. Whenever possible, one main result should be stated (or at least outlined) on the *first* page!
- Justifying why the main results are original and significant.
- Sketching or at least naming the methods used to obtain these results.
- Pointing out possible applications of the results and further directions of research.

- Outlining the structure of the paper.

In addition, the introduction should be accessible and attractive to a sufficiently large group of readers; hence, the prerequisites should be kept at a minimum. This contradicts the previous list of properties. Hence, it is impossible to write a perfect introduction. □

Fortunately, it is possible to write good approximations to a perfect introduction, i.e., writing an introduction without deviating from the ideal scenario in too many places. I recommend to write the introduction at a late stage of the writing process and to allow for enough time for rewriting the introduction as often as necessary.

4.9 Acknowledgements

A small paragraph with acknowledgements is in order if the paper was influenced by discussions or hints by colleagues, if the paper was supported through funding of a project, if an anomymous referee made helpful suggestions in a previous refereeing stage . . .

Typical locations for the acknowledgements are as a footnote on the first page, at the end of the introduction or at the end of the paper.

4.10 The abstract, again

Once the paper is completed, one should check whether the abstract still fits the paper and whether is still as good as one thought when first writing it. In many cases, it will be necessary to rewrite and improve the abstract at this point.

5 Layout

As a rule of thumb, one should not interfere with the layout – unless one knows exactly what one is doing. In general, the layouts of the standard LaTeX document classes are well-designed and adhere to typographical standards. These typographical standards have a long tradition and haven proved effective (e.g., that a line should contain about 60 to 80 symbols). Tempering with these things is dangerous and usually leads to bad results.

In journal publications, the journal dictates the layout. Sadly, many journal styles are not very convincing from the typographical point of view and terrible things might happen during production.

However, sometimes, layout is in your hands. A typical example is given by PhD theses. Some hints on improving the layout of your thesis are:

- Decide early on on the line width (other aspects of the layout can easily be adapted at a later stage).
- Use a proper binding correction.
- Before you get into a font frenzy: use as few fonts as possible and use matching math fonts.

6 Exercises, part I

Infrastructure

Exercise 6.1 (version control).

1. Choose your favourite version control system.
2. Initialise a repository for your next project.
3. Try out the basic commands of your version control system.

Exercise 6.2 (automated build process).

1. Install make (or something similar).
2. Write a Makefile that produces a pdf out of your LaTeXsource files.
3. Improve your Makefile to also automatically rerun pdflatex whenever necessary.

   Hints. Warning: This does not necessarily terminate! (Exercise 6.3)

Exercise 6.3 (LaTeX fixed points?!). Write a LaTeX document whose compilation does not have a fixed point (i.e., after each run of pdflatex, you would need to rerun pdflatex to get the correct resolution of references).

Exercise 6.4 (The Knuth Way).

1. Plan to write a comprehensive treatment of Computer Science.
2. Design a typesetting system for that purpose.
3. Use that typesetting system to write all your works.

Macros

Exercise 6.5 (useful macros).

1. Write a macro \comment that allows you to add meta-messages to your text (e.g., to remind you that certain parts still need to be written). It is recommended to use colour for such a command.
2. How can you easily check whether your document still contains calls to \comment?

   Hints. Let the compiler do the work!

3. Define a pagestyle (e.g, using the package scrpage2) that prints the compile date and time as well as the commit hash from your version control system on every page.

4. How can you easily achieve that these comments/date information do not show up in the document?

Exercise 6.6 (Heisenberg macros). Imagine that you write a paper on the Heisenberg group

$$\left\{ \begin{pmatrix} 1 & x & z \\ 0 & 1 & y \\ 0 & 0 & 1 \end{pmatrix} \;\middle|\; x, y, z \in \mathbb{Z} \right\} \subset \mathrm{SL}(3, \mathbb{Z}).$$

That means that you will have to type lots of such matrices ... Write a macro that makes this a pleasant experience!

Exercise 6.7 (PDE macros). Imagine that you write a paper on the *Estevez-Mansfield-Clarkson partial differential equation with Dirichlet boundary conditions*. Write a macro that saves a lot of writing in such a paper!

Exercise 6.8 (étale cohomology macros). Imagine that you write a paper on étale cohomology. Hence, you will have to type zillions of expressions of the form

$$H^k_{\text{ét}}\bigl(X, \mathbb{Z}/p^n(r)\bigr).$$

Write a macro that does most of the work for you! Which parts should be arguments of the macro and which will always be fixed? A tricky point is to make sure that the "ét" in the subscript is printed correctly also in theorem environments.

Exercise 6.9 (more macros).

1. Write a macro \lessonforlife that prints 225 times the sentence "Good macros save a lot of time and work!".

2. What does the macro \show do?

3. Look up what the commands \expandafter, \noexpand, \futurelet do.

Mathematical writing

Exercise 6.10 (local improvements). Improve/correct the formulations of the following theorems:

- Every differentiable function f is continuous.

- $f\colon V \longrightarrow V$ is a K-linear isomorphism if and only if $\det f \neq 0$.

- Let G be a group and let N be a subgroup. Then N is a normal subgroup of G if and only if

$$\forall_{g \in G} \quad \exists_{n' \in N} \quad g \cdot n = n' \cdot g \quad \forall_{n \in N}.$$

- Let K be a field and let V be a vector space. Then there exists a K-linear homomorphism $V \longrightarrow (V^*)^*$.

Exercise 6.11 (favourite theorem)**.** The goal of this exercise is to write a little exposition of a known theorem.

1. Choose your favourite theorem.

2. Define an audience for your text (students (specify the year)? researchers (specify the field)?).

3. State the theorem (keeping the guidelines from Section 3 in mind).

4. Make a list of terms and notation that is required to understand the theorem.

5. Write down a proof (or at least a sketch) of the theorem, paying attention to the following issues:
   - What is the main idea/strategy of the proof? Explain it!
   - Subdivide the proof into manageable steps.
   - Move auxiliary calculations etc. to separate lemmas.
   - Which external results do you need for the proof? Cite them appropriately!
   - Would it help to illustrate terms/arguments by figures?
   - Do you need to amend the list of terms/notation?

6. What is the original reference for your theorem? What is the reference for your proof? Cite these sources appropriately!

7. Make a list of applications of the theorem. Or at least provide an example in which the theorem is used.

8. Find another participant of the course and swap/discuss your texts!

Exercise 6.12 (dependencies)**.** Choose a research paper in your field and draw the dependency graph for all theorems/lemmas/intermediate steps. Would you organise the paper in a different way?

7 Exercises, part 2

More mathematical writing

Exercise 7.1 (abstracts)**.** Choose three research papers in your field and write new (better?) abstracts for them.

Exercise 7.2 (mini paper)**.** The goal of this exercise is to write a small research paper (for an audience that is familiar with your research area).

1. Choose a recent result of your own (no matter how small) and state this as the main theorem.
2. Choose a title for the paper.
3. Write a draft abstract for the paper.
4. Make a list of terms/notation that is required to understand the theorem, write down a proof (as in Exercise 6.11), make a list of terms/notation/other results that are needed to understand the proof the theorem.
5. Write a section collecting the background and preliminaries.
6. Polish the section(s) containing the main proof.
7. Add a section on applications.
8. Write the introduction.
9. Rewrite the abstract if necessary.
10. Add acknowledgements, keywords, MSC codes.
11. Apply a spell-checker to your text.
12. Reread your text two days after you completed it. Are you still happy with it? Rewrite your text if necessary.
13. Find another participant of the course and swap/discuss your texts!

Exercise 7.3 (random academic writing)**.** Write a program that randomly generates text that looks like a mathematical research paper (or at least a title or an abstract of such a paper). With some effort it is even possible to write a program that generates only mathematically correct texts. Make sure that the texts that you share with the scientific community have qualities that differ from such a randomly generated text!

Exercise 7.4 (further reading).

1. Read at laest one book on mathematical writing [23, 2, 13, 14, 15].
2. Read at least one book on Technical English [6, 24].
3. Read at least one book on typography [26, 27].

8 References

8.1 Examples used in the course

- M. Aigner, G.M. Ziegler. *Proofs from THE BOOK*, Springer, 1998.
  [p. 23]
- M. Alzenad, A. El-Keyi, F. Lagum, H. Yanikomeroglu. 3D Placement of an Unmanned Aerial Vehicle Base Station (UAV-BS) for Energy-Efficient Maximal Coverage, preprint,
  arXiv:1705.03415 [cs.IT], 2017.
  [first page]
- E. Dobrowolski. Casas-Alvero Conjecture is true, preprint,
  arXiv:1705.01704 [math.NT], 2017.
  [arXiv page]
- Y. Félix, S. Halperin, J.-C. Thomas. *Rational Homotopy Theory*, *Graduate Texts in Mathematics*, 205, Springer, 2001.
  [p. 29]
- M. Gromov. Volume and bounded cohomology, *Publ. Math. IHES*, 56, pp. 5–99, 1982.
  [p. 40]
- S. Hähne, J. Hornbostel. Semistable Symmetric Spectra in $\mathbb{A}^1$-homotopy theory, preprint, arXiv:1312.4340 [math.AT], 2013.
  [p. 30]
- C. Löh. Which finitely generated Abelian groups admit isomorphic Cayley graphs? *Geometriae Dedicata*, 164, pp. 97–111, 2013.
  [p. 100]
- J. Matoušek. *Thirty-three Miniatures: Mathematical and Algorithmic Applications of Linear Algebra*, *Student Mathematical Library*, 53, AMS, 2010.
  [p. 95]
- C. Neofytidis. Degrees of self-maps of products, preprint,
  arXiv:1512.03409v1 [math.AT], 2015.
  [p. 5]
- D. Plachky. *Wahrscheinlichkeitsrechnung, Diskrete Wahrscheinlichkeiten und Schätzen ihrer Parameter*, Oldenbourg, 1996.
  [p. 116]

- E. Rips, Y. Segev, Torsion-free group without unique product property, *Journal of Algebra*, 108(1), pp. 116–126, 1987.
  [p. 117]
- F. Waldhausen. Algebraic K-theory of spaces, *Algebraic and Geometric Topology (New Brunswick, N.J., 1983)*, *Lecture Notes in Mathematics*, 1126, pp. 318–419, Springer, 1985.
  [p. 348, as well as the mathscinet review]
- G.W. Whitehead. *Elements of Homotopy Theory*, *Graduate Texts in Mathematics*, 61, Springer, 1978.
  [p. 148, 149]

8.2 Literature

[1] *arXiv*
https://arxiv.org/ Cited on page: 3

[2] A. Beutelspacher. *Das ist o.B.d.A. trivial!*, 9th edition, Vieweg+Teubner, 2009. Cited on page: 19

[3] *The Coq Proof Assistant*
https://coq.inria.fr/ Cited on page: 3

[4] *darcs*
http://darcs.net/ Cited on page: 3

[5] *Detexify LaTeX handwritten symbol recognition*
http://detexify.kirelabs.org/classify.html Cited on page: 8

[6] L. Dupré. *Bugs in Writing: Guide to Debugging Your Prose*, Pearson Education, 1998. Cited on page: 9, 19

[7] *git*
https://git-scm.com/ Cited on page: 3

[8] *Google*
https://www.google.com/ Cited on page: 4

[9] *Mercurial*
https://www.mercurial-scm.org/ Cited on page: 3

[10] *The International Obfuscated C Code Contest*
http://www.ioccc.org/ Cited on page: 11

[11] D.E. Knuth. *The TeXbook*, Addison-Wesley, 1984. Cited on page: 7

[12] D.E. Knuth. *The METAFONT Book*, Addison-Wesley, 1986. Cited on page: 8

[13] D.E. Knuth, T.L. Larrabee, P.M. Roberts. *Mathematical Writing*, MAA, 1989. Cited on page: 9, 19

[14] L. Lamport. How to Write a Proof, *Amer. Math. Monthly*, 102(7), pp. 600–608, 1995. Cited on page: 11, 19

[15] L. Lamport. How to Write a 21st Century Proof, *Journal of Fixed Point Theory and Applications*, 11(1), pp. 43–63, 2012. Cited on page: 11, 19

[16] *latexmk – Fully automated LaTeX document generation* https://www.ctan.org/pkg/latexmk Cited on page: 3

[17] *GNU Make* https://www.gnu.org/software/make/ Cited on page: 3

[18] *mathoverflow* https://mathoverflow.net/ Cited on page: 4

[19] *MathSciNet* http://www.ams.org/mathscinet/index.html Cited on page: 4

[20] R. Murphy. *English Grammar in Use*, fourth edition, Cambridge University Press, 2012. Cited on page: 9

[21] *Oxford English Dictionary*, Oxford University Press http://www.oed.com/ Cited on page: 9

[22] M.H. Sørensen, P. Urzyczyn. *Lectures on the Curry-Howard Isomorphism*, *Studies in Logic and the Foundations of Mathematics*, 149, Elsevier, 2006. Cited on page: 3

[23] N.E. Steenrod, P.R. Halmos, M.M. Schiffer, J.A. Dieudonné. *How to Write Mathematics*, AMS, 1973. Cited on page: 19

[24] J.M. Swales, C.B. Feak. *Academic Writing for Graduate Students*, Michigan Series in English for Academic & Professional Purposes, The University of Michigan Press, 1994. Cited on page: 10, 19

[25] T. Tantau. *The* TikZ *and* PGF *Packages*, http://www.ctan.org/tex-archive/graphics/pgf/base/doc/generic/pgf/pgfmanual.pdf Cited on page: 8

[26] J. Tschichold. *Erfreuliche Drucksachen durch gute Typografie. Eine Fibel fr jedermann*, Maro Verlag, 2001. Cited on page: 19

[27] J. Tschichold. *Ausgewählte Aufsätze über Fragen der Gestalt des Buches und der Typographie*, zweite Auflage, Birkhäuser, 1987. Cited on page: 19

[28] *Wikipedia, The Free Encyclopedia* https://www.wikipedia.org/ Cited on page: 4

Clara Löh
clara.loeh@mathematik.uni-regensburg.de
Fakultät für Mathematik
Universität Regensburg
93040 Regensburg