Regular Lecture 2
Structure

Prof. Dr. Jasmin Blanchette

Chair of Theoretical Computer Science and Theorem Proving

Version of May 13, 2026

The Scaffolding

The Authors

The first question to answer is: Who should be an author?
For example, should your **supervisor** be a coauthor of your research paper?

According to the German Research Foundation, "an author is an individual who has made a genuine, identifiable contribution to the content of a research publication of text, data or software."[1]

[1] German Research Foundation, *Guidelines for Safeguarding Good Research Practice: Code of Conduct*, 2019.

The Authors

The order of authors is also a concern. Three common approaches:

- **alphabetical** by family name;
- by **decreasing contributions**, starting with the main contributor or contributors;
- by **increasing seniority**, starting with the most junior author.

The most visible position is the first (cf. *Zhang et al.*).
Junior authors typically need the visibility more than their senior colleagues.
The last position, identifying the "**head honcho**," is also coveted.

The Authors

Roles of authors:

- The "**organizer**" coordinates the work, keeping an eye on submission venues and deadlines.
- The "**editor**" polishes the writing and unifies the style.
- The **corresponding author** is the main point of contact for the publisher and readers.
- The **main contributors** carry out most of the work.

For a single-author document, the same person performs all these tasks.

The Topic

Often different authors have different ideas about the topic.
It helps clarify the following at the beginning and to have **all authors agree** on them:

- tentative title;
- tentative abstract;
- tentative table of contents;

and, for a thesis or research paper:

- tentative list of contributions.

These will keep evolving until the manuscript is finalized.

The Perspective

Is your text **objective** or **subjective**?

A thesis or research paper will try to convey that its contributions are important.
Otherwise, the thesis might get a poor grade, or the paper might get rejected.

Even software documentation can be subjective.
Suppose you work for a company and write documentation for its product.
Since the documentation is a form of advertisement for the product,
you will not go to great lengths to emphasize bugs.

Some authors use the passive voice to avoid *I* and *we*,
but this alone does not make your ideas more objective.
Conversely, use *I* only if you are relevant to the story.

The Type of Writing

Narrative writing

The narrative style tells a story, commonly in the order in which it unfolds. Fiction and biographies often use this style.

Expository writing

Most scientific and technical writing is expository:
It seeks to convey information.

Persuasive writing

Opinion pieces, advertisement but also grant proposals and customer support materials aim to persuade. Following an antique tradition, you can break down argumentative texts into **ethos**, **logos**, and **pathos**: You establish your credibility, make your case, and end with an appeal to emotions.

The Target Audience

Your primary **responsibility is to readers**, not reviewers, colleagues, or even yourself.

For seminar papers and most theses, the audience is largely imaginary—hardly anybody except your instructor or supervisor will read them. But even in such an artificial setting, you should write with an audience in mind (e.g., fellow students).

The Title

Titles should ideally be both **informative** and **memorable**. Two popular approaches:

- A **catchy title** arouses the reader's curiosity—e.g., *We Know (Nearly) Nothing! But Can We Learn?*[1]
- An **informative title** succinctly captures the document's content—e.g., *Combining Inductive and Deductive Techniques in Proof Search.*

Each approach has its advocates.

A **double-barreled title** combines a catchy title with an informative subtitle—e.g., *Data for Good: Ensuring the Responsible Use of Data to Benefit Society*[2]

Choose a tentative title early. Then keep reworking it.

[1] Stephan Schulz, *ARCADE 2017*, pp. 29–32, EasyChair, 2017.
[2] Jeannette M. Wing, *BigData 2020*, pp. 1–2, IEEE, 2020.

Examples of Titles

Succinct

Local Type Inference[1]
Typeful Programming[2]

Flowery

Synchronizing the Asynchronous[3]
Ornamental Algebras, Algebraic Ornaments[4]

---

[1] Benjamin C. Pierce and David N. Turner, *ACM Transactions on Programming Languages and Systems* 22(1), pp. 1–44, 2000.

[2] Luca Cardelli, *Formal Description of Programming Concepts*, pp. 431–507, Springer, 1989.

[3] Bernhard Kragl, Shaz Qadeer, and Thomas A. Henzinger, *CONCUR 2018*, pp. 21:1-21:17, Schloss Dagstuhl, 2018.

[4] Conor McBride, unpublished manuscript, 2010.

Examples of Titles

Wordplay

Time for Logic[1]

Decoding Choice Encodings[2]

Double-barreled

Anytime, Anywhere: Modal Logics for Mobile Ambients[3]

The $25,000,000,000 Eigenvector: The Linear Algebra Behind Google[4]

---

[1] Rajeev Alur and Thomas A. Henzinger, "SIGACT News Logic Column 1: Time for Logic," *ACM SIGACT News* 22(3), pp. :6–12, 1991.

[2] Uwe Nestmann and Benjamin C. Pierce, *Information and Computation* 163(1), pp. 1–59, 2000.

[3] Luca Cardelli and Andrew D. Gordon, *POPL 2000*, pp. 365–377, ACM, 2000.

[4] Kurt Bryan and Tanya Leise, *SIAM Review* 48(3), pp. 569–581, 2006.

Examples of Titles

Humorous

All Your IFCException Are Belong to Us[1]

Functional Programming with Bananas, Lenses, Envelopes and Barbed Wire[2]

k + *1 Heads Are Better than* k[3]

Questioning

What is Decidable about Gradual Types?[4]

Would Turing Have Won the Turing Award?[5]

---

[1] Cătălin Hriţcu, Michael Greenberg, Ben Karel, Benjamin C. Pierce, and Greg Morrisett, *SP 2013*, pp. 3–17, IEEE, 2013.

[2] Erik Meijer, Maarten M. Fokkinga, and Ross Paterson, *FPCA 1991*, pp. 124–144, Springer, 1991.

[3] Andrew C. Yao and Ronald L. Rivest, *Journal of the ACM* 25(2), pp. 337–340, 1978.

[4] Zeina Migeed and Jens Palsberg, *Proceedings of the ACM on Programming Languages* 4(POPL), pp. 29:1-29:29, 2020.

[5] Moshe Y. Vardi, *Communications of the ACM* 60(11), p. 7, 2017.

The Abstract

Most theses and papers have an abstract or summary.
The role of the abstract is to allow potential readers in the target audience to quickly determine whether they want to read the rest.

Usual format:

- brief statement of the **context** in which the work fits or the question it answers;
- brief statement of the **main idea**;
- brief summary of the **contributions**.

The Abstract

Early on, perhaps months before your (real or imaginary) deadline,
write a tentative abstract. This will help **crystallize your thoughts**.

The best abstracts are **short**. For a 15-page paper, four sentences may suffice.
The introduction will provide more details anyway.

The abstract is **meta-information**. Readers might skip it.
Everything you mention in it should reappear with more detail in the text's body.
In particular, any concept, notation, or abbreviation introduced in the abstract
must be reintroduced later.

The Table of Contents

Early on, create a rough table of contents.

In which order should the material be covered?

- Some **dependencies** arise naturally and should be honored.
- There is also the **seductive order**: Pull the most interesting material to the front, and push the tedious details to the end.

In your draft, annotate the table of contents with **page budgets** and, for documents with multiple authors, **author initials** indicating who is responsible for writing what.

Example of a Draft Table of Contents

Title and Abstract	*0.5*	
1. Introduction	*1.5*	*TS*
2. Background	*2.5*	*EF+MF*
2.1. Quality Estimation	*(1.5)*	*MF*
2.2. Learning	*(1)*	*EF*
3. Data Set	*2*	*NK*
4. Experimental Setup	*3*	*MF+NK*
4.1. Study Design	*(1.5)*	*NK*
4.2. Method and Procedure	*(1.5)*	*MF*
5. Results	*1*	*NK*
6. Discussion	*0.75*	*TS*
7. Conclusion	*0.5*	*TS*
Acknowledgment	*0.25*	
	12	

The Page Budget

One way to come up with a page budget is to take your page count target, divide it by how many chapters or sections you plan, and **adjust the numbers**.

Take into account that the conclusion is usually short, and the introduction and background should typically represent no more than 25% of your document. The reader then quickly reaches the document's **core** chapters or sections.

Within the core, **expand** the interesting material, and **compress** the tedious details (or relocate them to an appendix or a technical report).

The Writing Process

Gathering Ideas

Create an **online document** or **repository** early on in which
you and your coauthors can collect ideas.
Whenever you have an idea, write it down using your favorite approach
(e.g., bullet points, mind maps, Zettelkasten).

Start thinking about **key terminology** and **notations**.
For mathematical texts, decide which letters stand for which concepts
(e.g., x for real numbers, M for Turing machines).

Also keep track of everybody who helped you,
so that you remember to **acknowledge** them later.

From Ideas to Sketch

Next, enrich your document with more

- **bullet points**;
- **examples**;
- **tables**;
- **figures**;
- etc.

Then put the items in order. Group them into (what will become) paragraphs. Each paragraph should contain one idea. Break long paragraphs along subideas.

Use this sketch as a basis for further thought and discussions with coauthors.

From Sketch to Text

Once your sketch is sufficiently developed, you can use it to write text that respects the **norms** and **conventions** of the target genre.

You can write the text in any order.

- For example, you can collect ideas for Section 10 while sketching Section 7 while writing Section 2.
- If you have coauthors, they can also **work in parallel** on different sections.

The sketch ensures basic coherence.

With this simple approach, you can defeat the **blank page syndrome**.

The Curse of Knowledge

As you write, remember that you suffer from the curse of knowledge:
You might not notice that you rely on **specialized knowledge** or **jargon**.

According to Pinker, "the main cause of incomprehensible prose is the difficulty of imagining what it's like for someone else not to know something that you know."

Knuth made a similar observation: "If you try to write for the novice, you will communicate with the experts—otherwise you communicate with nobody."

The Curse of Knowledge

Symptoms:

- omitting essential background information;
- not defining unfamiliar terms;
- writing too abstractly, without examples;
- omitting logical links and jumping to conclusions.

Solutions:

- **Keep checking** for the above symptoms.
- **Wait a few days or weeks**, until the text is no longer familiar, and reread it.
- Reread the text while **imagining yourself** to be a member of your target audience.
- **Get feedback** from your target audience.

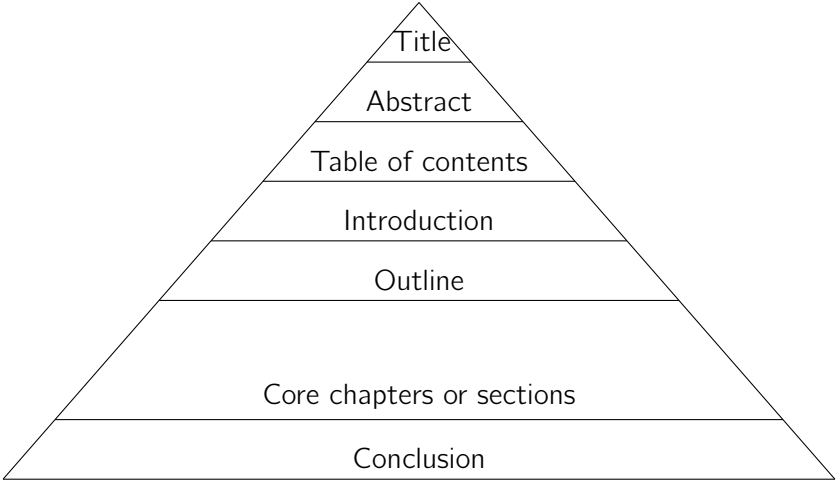
The Pyramid Principle

Structure of Well-Structured Texts

Well-structured texts follow the "**cascading level of detail**" approach:

1. The title very succinctly summarizes the entire document.
2. The abstract, if present, constitutes a summary.
3. The table of contents is itself a summary.
4. The introduction provides a more detailed summary.
5. The outline at the end of the introduction, if present, lists the remaining chapters or sections, a form of summary.
6. The core chapters or sections present the same ideas in greater detail.
7. The conclusion is yet another summary—one that recapitulates.

The Pyramid Principle Illustrated

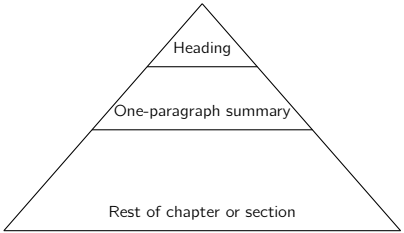


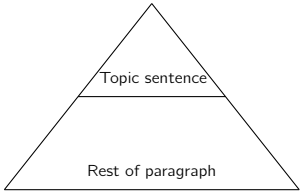
Pyramids All the Way Down

The pyramid principle applies **recursively**, like nested dolls:

- Each of the core chapters or sections may start with a **paragraph** that summarizes it.
- Each paragraph may start with a **topic sentence** that summarizes it.

Respectively:





Advantages of the Pyramid Principle

With the pyramid principle:

- The reader quickly understands **where the text is going**.
- If the reader stops reading after the title, abstract, or introduction, they will have a reasonable **understanding of the whole**.
- A "**grasshopper**" reader can exploit the structure to skim the text, skipping entire chapters, sections, or paragraphs without losing the thread.
- The redundancy naturally **emphasizes the essential points**.

Discover the Pyramids

Let us analyze the structure of two research papers:

- Xuejun Yang, Yang Chen, Eric Eide, and John Regehr, "**Finding and Understanding Bugs in C Compilers**," *PLDI 2011*, pp. 283–294, ACM, 2011.
- Christel Baier and Marta Kwiatkowska, "**Domain Equations for Probabilistic Processes**," *Mathematical Structures in Computer Science* 10(6), pp. 665–717, 2000.