

How to write a paper

How to write a ~~paper~~  
technical report

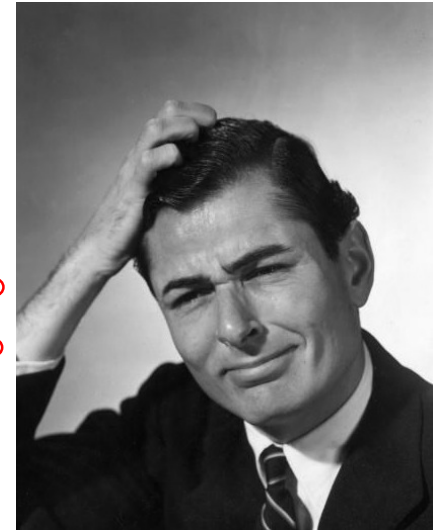
# Why start with a presentation?



- Because it helps us focus on the main points!

What's the problem?

Why should I care?



# An Example: "Good Examples"

```
V = spawn()      -> '<0.27813.1>',  
register(a, V)   -> true,  
whereis(a)      -> '<0.27813.1>'
```

```
V = spawn()      -> '<0.27164.1>',  
register(b, V)   -> true,  
register(a, V)   -> {'EXIT', badarg}
```

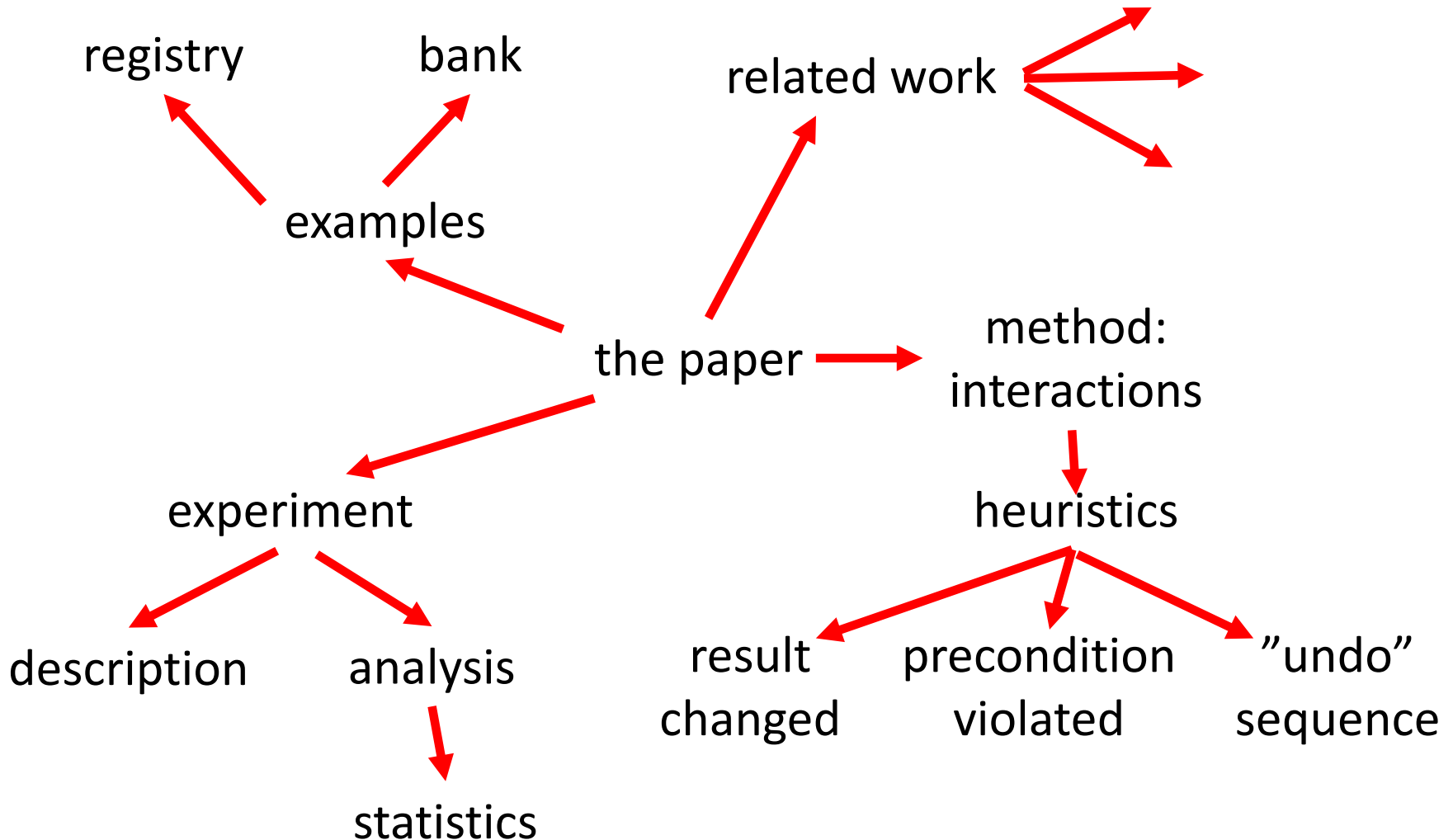
```
V1 = spawn()     -> '<0.28614.1>',  
V2 = spawn()     -> '<0.28615.1>',  
register(c, V1)  -> true,  
register(c, V2)  -> {'EXIT', badarg}
```

## **ABSTRACT**

Formal specifications of software applications are hard to understand, even for domain experts. Because a formal specification is abstract, reading it does not immediately convey the expected behaviour of the software. Carefully chosen examples of the software's behaviour, on the other hand, are concrete and easy to understand—but poorly-chosen examples are more confusing than helpful. In order to understand formal specifications, software developers need *good examples*.

We have created a method that automatically derives a suite of good examples from a formal specification. Each example is judged by our method to illustrate one feature of the specification. The generated examples give users a good understanding of the behaviour of the software. We evaluated our method by measuring how well students understood an API when given different sets of examples; the students given our examples showed significantly better understanding.

# What should go into the paper?



# Structure

- Abstract
- Introduction
- Background
- Main idea
- Details
- Evaluation
- Related work
- Discussion/Conclusion
- References



# Structure

- Abstract
- Introduction
- Background
- Main idea
- Details
- Evaluation
- Related work
- Discussion/Conclusion
- References

## **Abstract**

- A *brief* summary: what's in the paper?
- Write this last!

# Structure

- Abstract
- Introduction
- Background
- Main idea
- Details
- Evaluation
- Related work
- Discussion/Conclusion
- References



## **Introduction**

- What is the problem?
- Why is it important?
- Give an example!

# Structure

- Abstract
- Introduction
- Background
- Main idea
- Details
- Evaluation
- Related work
- Discussion/Conclusion
- References

## Background

- Things you really have to explain *before* the key idea
- Keep this short, or you lose people! The reader must already be interested.
- (Preferably introduce background where it is used)

# Structure

- Abstract
- Introduction
- Background
- Main idea
- Details
- Evaluation
- Related work
- Discussion/Conclusion
- References

## **Main idea**

- The key idea in its simplest form
- Give examples!
- Make sure reader understands the gist

# Structure

- Abstract
- Introduction
- Background
- Main idea
- Details
- Evaluation
- Related work
- Discussion/Conclusion
- References

## **Details**

- Technical details, handling tricky cases, etc
- Proofs if necessary
- Enable the reader to reconstruct the work

# Structure

- Abstract
- Introduction
- Background
- Main idea
- Details
- Evaluation
- Related work
- Discussion/Conclusion
- References

## **Evaluation**

- Why should we believe this is a good idea?
- How have you shown that it works?
- Experimental results, applications, etc

# Structure

- Abstract
- Introduction
- Background
- Main idea
- Details
- Evaluation
- Related work
- Discussion/Conclusion
- References

## **Related work**

- How have the same (or similar) problems been addressed before?
- How does this solution compare?
- Placed towards the end to permit detailed comparisons

# Structure

- Abstract
- Introduction
- Background
- Main idea
- Details
- Evaluation
- Related work
- Discussion/Conclusion
- References

## **Discussion/Conclusion**

- What have we shown?
- What are the natural next steps ("future work")?



# Structure

- Abstract
- Introduction
- Background
- Main idea
- Details
- Evaluation
- Related work
- Discussion/Conclusion
- References



## **References**

- Papers you refer to



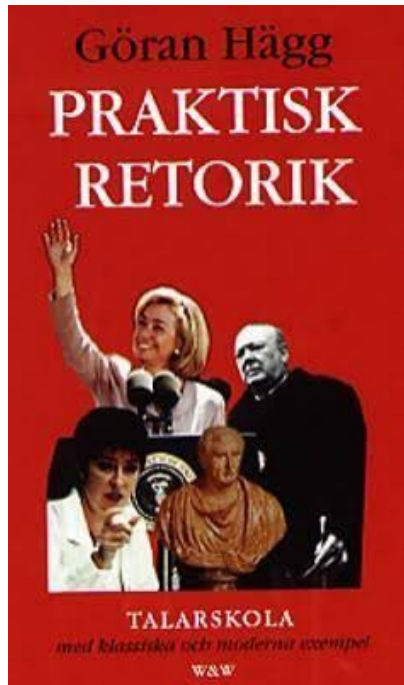
## Structure (conference paper)

---

- Title (1000 readers)
- Abstract (4 sentences, 100 readers)
- Introduction (1 page, 100 readers)
- The problem (1 page, 10 readers)
- My idea (2 pages, 10 readers)
- The details (5 pages, 3 readers)
- Related work (1-2 pages, 10 readers)
- Conclusions and further work (0.5 pages)

# A structure that works

- **Abstract** (1-2 paragraphs, 1000 readers)
- **Intro** (1-2 pages, 100 readers)
- **Main ideas** (2-3 pages, 50 readers)
- **Technical meat** (4-6 pages, 5 readers)
- **Related work** (1-2 pages, 100 readers)



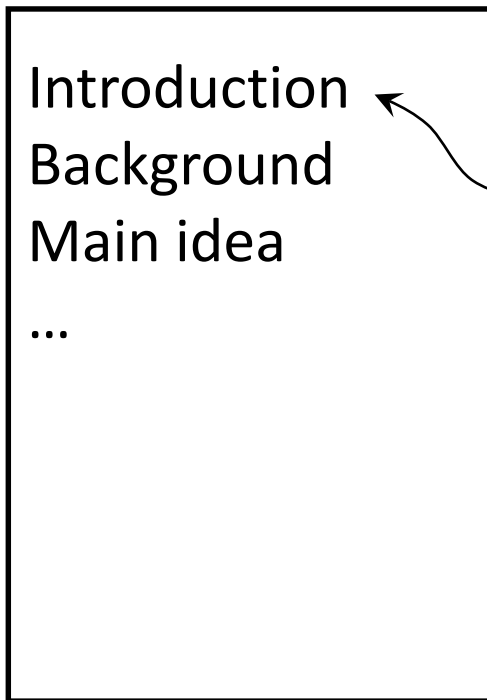
- Exordium (Hello)
- Narratio (Background)
- Probatio (My idea)
- Refutatio (Related work?)
- Peroratio (Conclusion)

# “Good Examples” paper

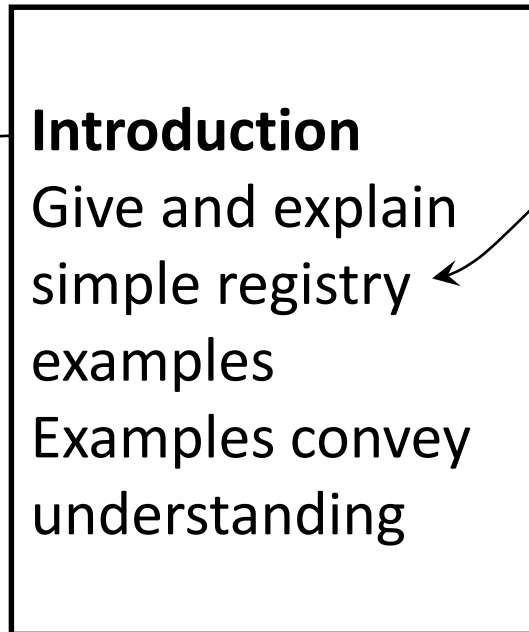
- Abstract
  - A Running Example
  - The Question
  - State Machine Models
  - The Idea: Finding Interactions
  - Precondition Interactions
  - Extensions
  - Evaluation
  - Comparison to Related Work
  - Conclusion
- We started with an example!
- We pose the problem
- Necessary background (1/2 page)
- The main idea in its simplest form
- Details
-

# Writing Process

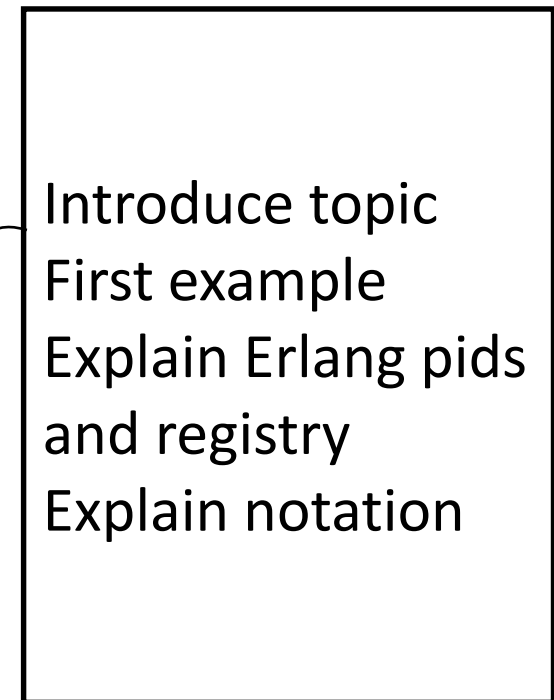
- Plan the sections



- Plan each section—notes and keywords



- Plan each paragraph



# Then write

We explained a 'pid'

What's Erlang?

## 1 A RUNNING EXAMPLE

Let us start by introducing our running example—the Erlang program. Erlang is a concurrent programming language in which a program is made up of very many lightweight processes, each with its own process identifier, or *pid*. For illustration, we shall create (dummy) processes using a function `spawn()`, which returns the created process' pid as its result. To enable processes to find each other, they may be *registered* under a name, names being Erlang atoms, which are similar to strings in many other languages. For example, we might create and register a process like this:

```
V = spawn()      -> '<0.27813.1>',  
register(a, V)   -> true,  
whereis(a)      -> '<0.27813.1>'
```

We introduced spawn

We explained names

```
V = spawn()      -> '<0.27813.1>',  
register(a, V)   -> true,  
whereis(a)      -> '<0.27813.1>'
```

We explain  
the notation

Here we show the arguments and results of a sequence of calls in a sample run; '`<0.27813.1>`' is an example of a dynamically created pid, and the atom `a` is the name assigned to the pid in this case. We indicate reuse of a result by binding it to a variable: the result of `spawn` is bound to the variable `V`, then passed as an argument to `register`. Finally, `whereis` looks up a name in the registry and returns the associated pid.

We  
explain  
"V="

We explain  
what this  
example shows





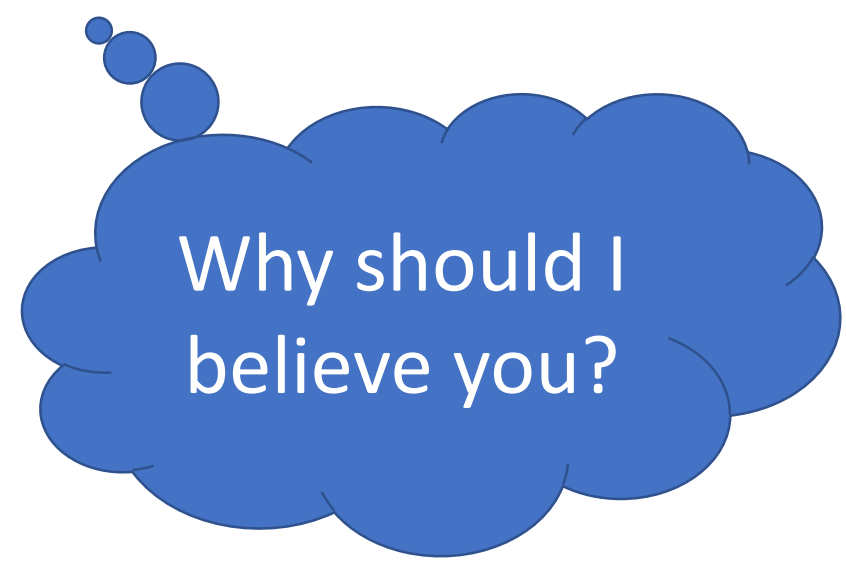
We hope that, thanks to these examples, the reader now has a good understanding of the behaviour of this simple API. The point, though, is that while the words above were written by the authors, *the examples were generated and chosen automatically* from a formal specification of the API, using the methods that are the subject of this paper.

Examples, examples, examples

Don't write in abstract  
generalities!



What do you  
*mean?*



Why should I  
believe you?

Where should I start writing?



What does the reader know?

# Iteration

- Introduction



- Background



- Main idea



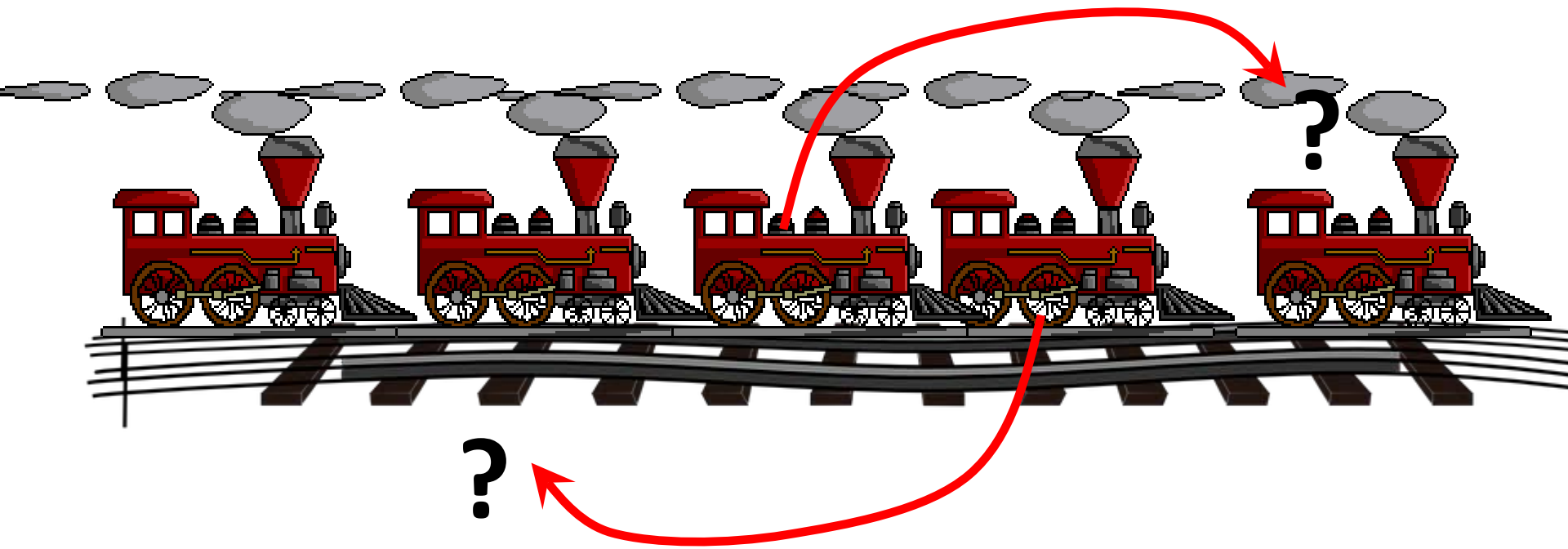
- Details

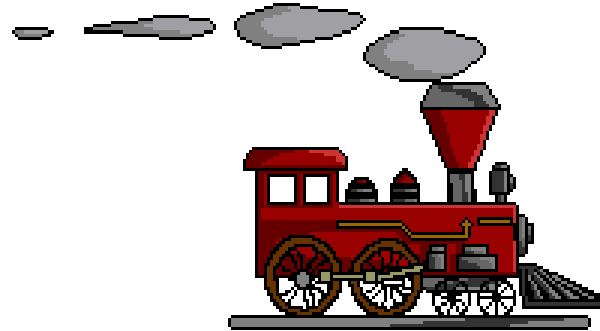
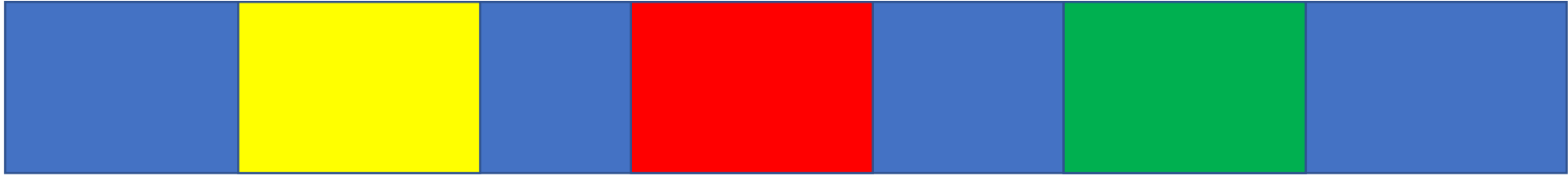


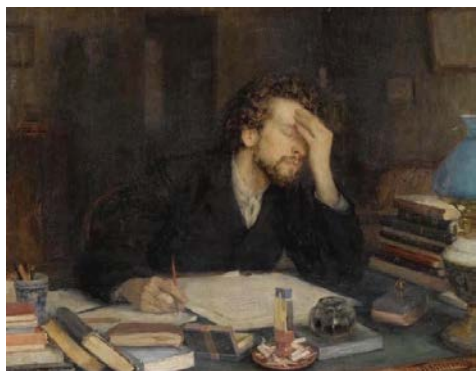
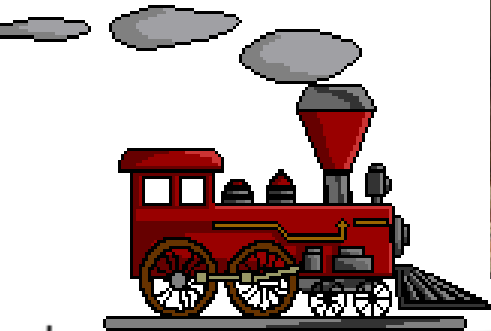
- ...



Revision







**As we will be  
explaining in  
section 4, ...**

*As we already  
explained in  
section 2, ...*



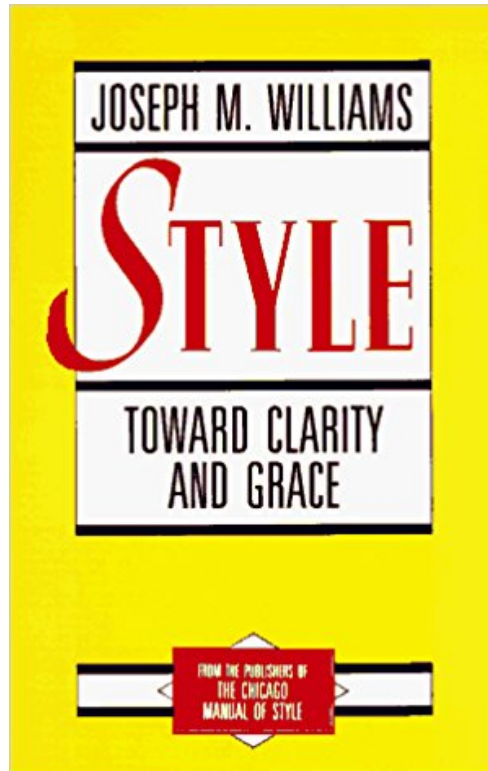
**As we will be  
explaining in  
section 4, ...**

**As we already  
explained in  
section 2, ...**

After the first draft, *everyone owns the entire text.*



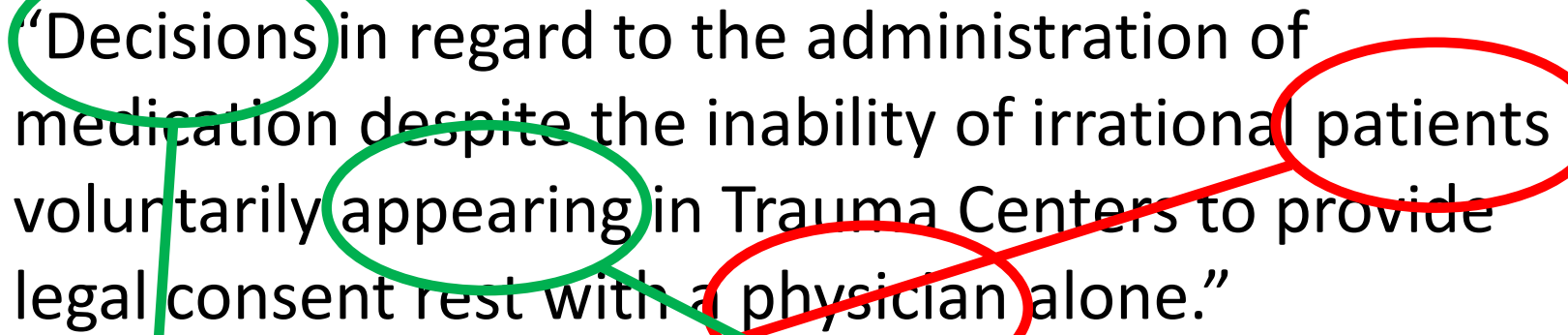
NEVER  
RE-USE  
TEXT!!!



- What's wrong with a sentence or paragraph
- *How to fix it!*

# Clarity

“Decisions in regard to the administration of medication despite the inability of irrational patients voluntarily appearing in Trauma Centers to provide legal consent rest with a physician alone.”



“When a patient voluntarily appears at a Trauma Center but behaves so irrationally that he cannot legally consent to treatment, only a physician can decide whether to administer medication.”

***Who did what?***

# Clarity

- Who are the main characters?
  - Make them the *subjects* of sentences
- What are the main actions?
  - Make them the *verbs* in sentences

# Cohesion: does it flow?

“Security proofs of cryptographic protocols are crucial for the security of everyday electronic communication. However, these proofs tend to be complex and difficult to get right. The game-playing technique, originally proposed by Jones et al., follows a code-based approach where the security properties are formulated in terms of probabilistic programs, called games. This is a general design principle for cryptographic proofs to ease their management.”

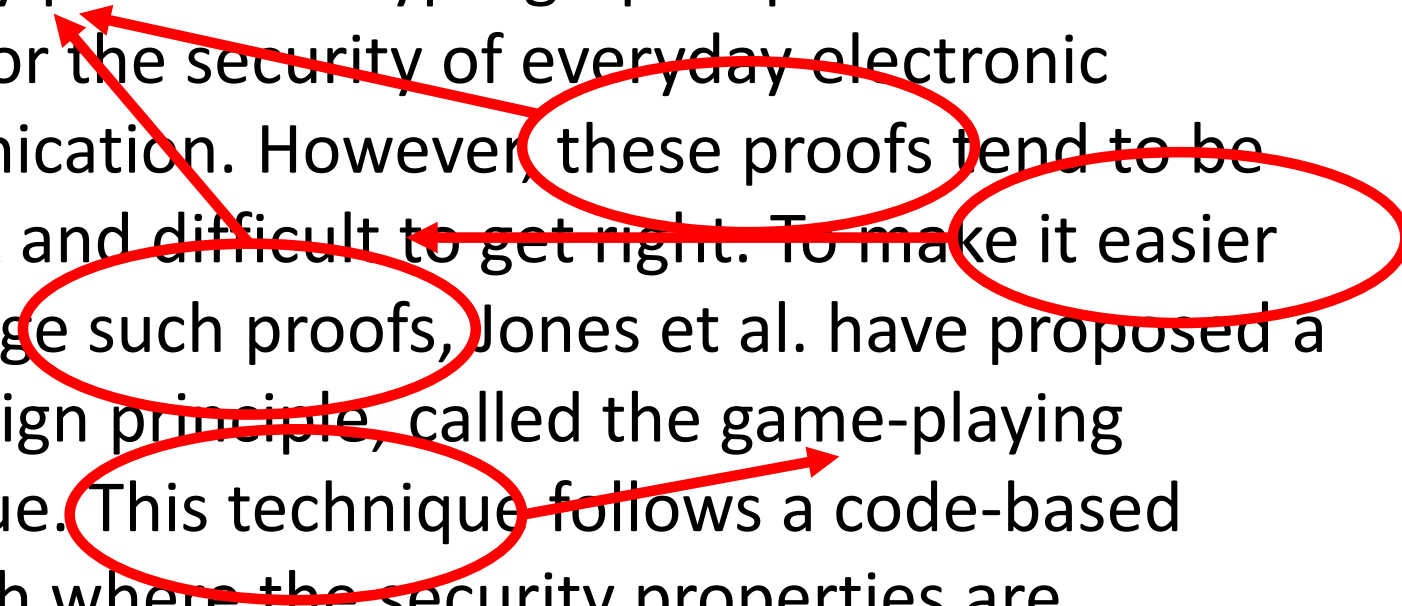
# Cohesion: does it flow?

“Security proofs of cryptographic protocols are crucial for the security of everyday electronic communication. However, these proofs tend to be complex and difficult to get right. **The game-playing technique, originally proposed by Jones et al., follows a code-based approach where the security properties are formulated in terms of probabilistic programs, called games.** This is a general design principle for cryptographic proofs to ease their management.”

# Flow

- Begin sentences with old information
  - Create link to what came before
- Place *new* information at the end of a sentence
  - Create link to what follows
  - Puts new info in a position of emphasis

“Security proofs of cryptographic protocols are crucial for the security of everyday electronic communication. However, these proofs tend to be complex and difficult to get right. To make it easier to manage such proofs, Jones et al. have proposed a new design principle, called the game-playing technique. This technique follows a code-based approach where the security properties are formulated in terms of probabilistic programs, called games.”

The image contains several red annotations on the text. There are five red ovals highlighting the following phrases: "these proofs tend to be", "complex and difficult to get right.", "To make it easier", "such proofs,", and "This technique". Additionally, there are four red arrows: one pointing from the first oval to "Security proofs", one from the second oval to "difficult", one from the third oval to "easier", and one from the fourth oval to "follows".



# Cohesion: topics

”Particular ideas toward the beginning of each clause define what a passage is centrally "about" for a reader, so a sense of coherence crucially depends on topics. Cumulatively, the thematic signposts that are provided by these ideas should focus reader’s attention toward a well-defined and limited set of connected ideas. Moving through a paragraph from a cumulatively coherent point of view is made possible by a sequence of topics that seem to constitute this coherent sequence of topicalized ideas. A seeming absence of context for each sentence is one consequence of making random , shifts in topics. Feelings of dislocation, disorientation, and lack of focus will occur when that happens.”

# Cohesion: topics

”Particular ideas toward the beginning of each clause define what a passage is centrally "about" for a reader, so a sense of coherence crucially depends on topics. Cumulatively, the thematic signposts that are provided by these ideas should focus reader’s attention toward a well-defined and limited set of connected ideas. Moving through a paragraph from a cumulatively coherent point of view is made possible by a sequence of topics that seem to constitute this coherent sequence of topicalized ideas. A seeming absence of context for each sentence is one consequence of making random , shifts in topics. Feelings of dislocation, disorientation, and lack of focus will occur when that happens.”

# Cohesion: topics

”Particular ideas toward the beginning of each clause define what a passage is centrally "about" for a reader, so a sense of coherence crucially depends on topics. Cumulatively, the thematic signposts that are provided by these ideas should focus reader’s attention toward a well-defined and limited set of connected ideas. Moving through a paragraph from a cumulatively coherent point of view is made possible by a sequence of topics that seem to constitute this coherent sequence of topicalized ideas. A seeming absence of context for each sentence is one consequence of making random , shifts in topics. Feelings of dislocation, disorientation, and lack of focus will occur when that happens.”

# Cohesion: topics

”Particular ideas toward the beginning of each clause define what a passage is centrally "about" for a reader, so a sense of coherence crucially depends on topics. Cumulatively, the thematic signposts that are provided by these ideas should focus reader’s attention toward a well-defined and limited set of connected ideas. Moving through a paragraph from a cumulatively coherent point of view is made possible by a sequence of topics that seem to constitute this coherent sequence of topicalized ideas. A seeming absence of context for each sentence is one consequence of making random , shifts in topics. Feelings of dislocation, disorientation, and lack of focus will occur when that happens.”

# Cohesion: topics

”**Particular ideas** toward the beginning of each clause define what a passage is centrally "about" for a reader, so a sense of coherence crucially depends on topics. Cumulatively, the **thematic signposts** that are provided by these ideas should focus reader’s attention toward a well-defined and limited set of connected ideas. **Moving through a paragraph** from a cumulatively coherent point of view is made possible by a sequence of topics that seem to constitute this coherent sequence of topicalized ideas. A **seeming absence of context** for each sentence is one consequence of making random , shifts in topics. Feelings of dislocation, disorientation, and lack of focus will occur when that happens.”

# Cohesion: topics



Confusing topic string!

”**Particular ideas** toward the beginning of each clause define what a passage is centrally "about" for a reader, so a sense of coherence crucially depends on topics. Cumulatively, the **thematic signposts** that are provided by these ideas should focus reader’s attention toward a well-defined and limited set of connected ideas. **Moving through a paragraph** from a cumulatively coherent point of view is made possible by a sequence of topics that seem to constitute this coherent sequence of topicalized ideas. A **seeming absence of context** for each sentence is one consequence of making random , shifts in topics. **Feelings of dislocation**, disorientation, and lack of focus will occur when that happens.”

# Cohesion: topics

“**Topics** are crucial for a reader because they focus the reader's attention on a particular idea toward the beginning of a clause and thereby notify a reader what a clause is ‘about.’ **Topics** thereby crucially determine whether the **reader** will feel a passage is coherent. Cumulatively, through a series of sentences, these **topicalized ideas** provide thematic signposts that focus the reader’s attention on a well-defined set of connected ideas. If a **sequence of topics** seems coherent, that consistent sequence will move the reader through a paragraph from a cumulatively coherent point of view. But if through that paragraph **topics** shift randomly, then the **reader** has to begin each sentence out of context, from no coherent point of view. When that happens, the **reader** will feel dislocated, disoriented, out of focus.”

# Emphasis

“No one can explain why that first primeval superatom exploded and thereby created the universe in a few words.”



# Emphasis

“No one can explain why that first primeval superatom exploded and thereby created the universe **in a few words.**”



“No one can explain in a few words why that first primeval superatom exploded and thereby **created the universe.**”

*Moving the important information to the end of a sentence is another way to manage the flow of ideas.*



*Another way you can manage the flow of ideas is to move the most important information to the **end of the sentence.***



We hope that, thanks to these examples, the reader now has a good understanding of the behaviour of this simple API. The point, though, is that while the words above were written by the authors, *the examples were generated and chosen automatically* from a formal specification of the API, using the methods that are the subject of this paper.

# Coherence

“Lions and tigers are some of the most majestic and awe-inspiring species of cats. Most of these big cats, however, are currently facing extinction. There has been more evolutionary success in smaller cats. Although house cats are considered the most popular pet in the world, they are in fact the most common. It would therefore be interesting to see if house cats can be trained to be more social.

What is the point?

# Coherence

- It should be clear how each sentence relates to the *big picture*.
- A paragraph should have *one main point*, expressed in a *point sentence* at (or near) the beginning.

# Coherence: point sentence at the beginning

“There appears to be a negative correlation between the charisma of a species and its ability to survive.

Lions and tigers, for instance, are among the most majestic creatures in the animal kingdom, yet they are currently facing extinction. In contrast, the house cat is evolutionarily quite successful, even though it is mostly known for stupid pet tricks.”

# Concision

“The point I want to make here is that we can see that American policy in regard to foreign countries as the State Department in Washington and the White House have put it together and made it public to the world has given material and moral support to too many foreign factions in other countries that have controlled power and have then had to give up the power to other factions that have defeated them.”



“Our foreign policy has backed too many losers.”

# Concision

consider

~~“In my personal opinion, we must listen to and think~~  
~~over in a punctilious manner each and every~~  
suggestion ~~that is offered to us.”~~

We must consider each suggestion carefully.

Any revision that shortens a sentence is likely to be a good one.

You *will* write long, unwieldy sentences. We all do!  
Reading and re-reading reveals them.



# Use names consistently

“John is giving a lecture. The Professor is using the projector to show his slides. Hughes thinks nomenclature is important.”

# References

...

This insight has been described more appropriately by **Hughes**, Thompson, and surely others [19,33].

...

## References

[1] L. Augusteijn, Sorting morphisms, in: S. Swierstra, P. Henriques, J. Oliveira (Eds.), ...

...

[19] J. **Hughes**, Why functional programming matters, The Computer Journal 32 (2) (1989) 98–107.

## LaTeX

...

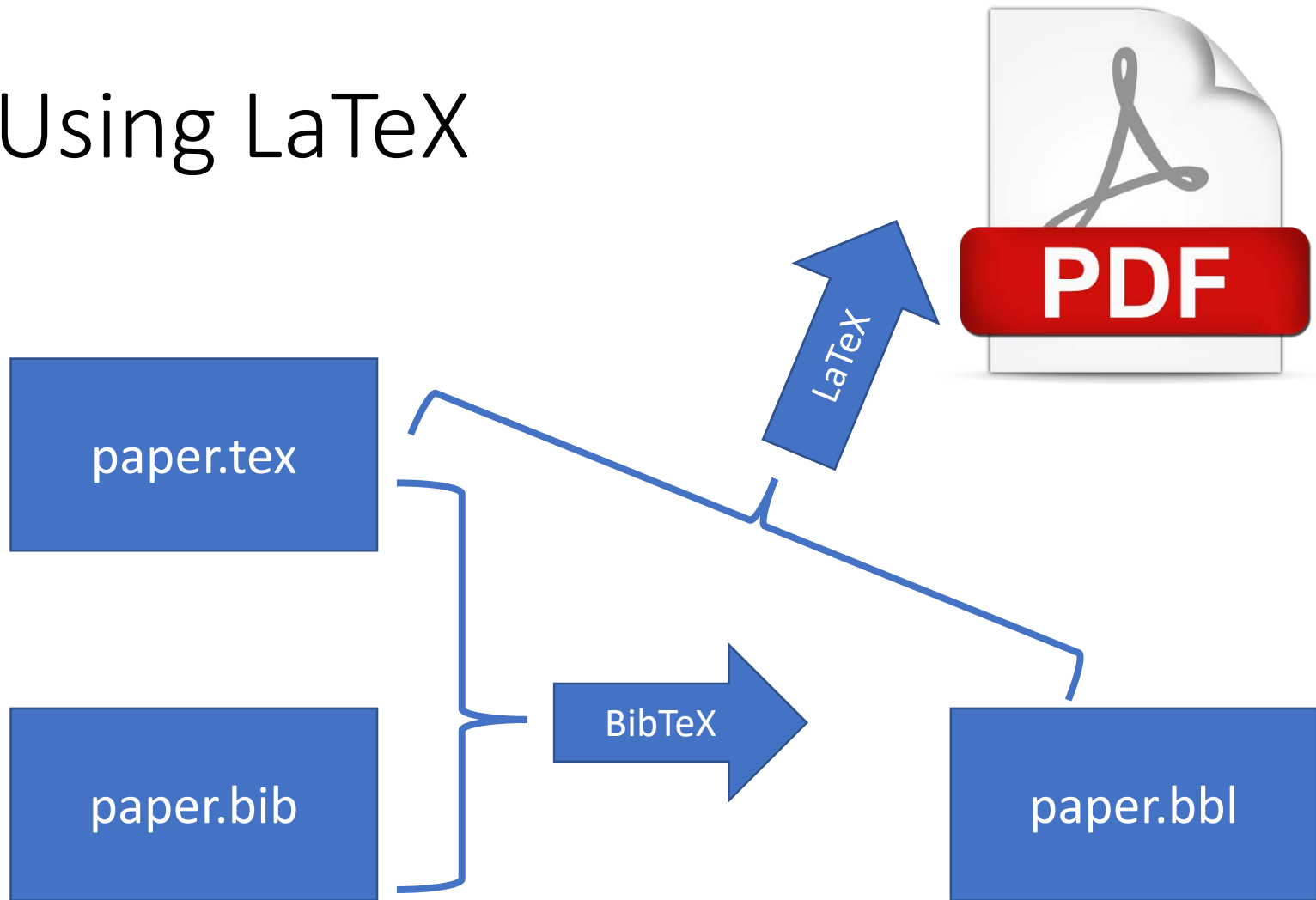
This insight has been described more appropriately by **Hughes**, Thompson, and surely others  
`\cite{hughes,thompson}`.

...

## BibTeX

```
@incollection{hughes,  
  author = {Hughes, John},  
  chapter = {Why Functional  
            Programming Matters},  
  title = {Research Topics in Functional  
          Programming},  
  editor = {Turner, David A.},  
  year = {1990}}
```

# Using LaTeX



# Where do BibTeX entries come from?

Why functional programi X

Säker | <https://dl.acm.org/citation.cfm?id=119832>

Appar Thermia Online Wales Tour 2017 - Mi Don's Synology TimeEdit Chalmers Quiviq issue tracker Product videos Polar

ACM DL DIGITAL LIBRARY UPMC Universite Pierre et Marie CURIE My Binders SIGN OUT: Prof John Hughes

Export Formats

BibTeX | EndNote | ACM Ref

Why fun

Author:

Published

Book

Research

Pages 1

Addison-W

table of con

```
@incollection{Hughes:1990:WFP:119830.119832,
author = {Hughes, John},
chapter = {Why Functional Programming Matters},
title = {Research Topics in Functional Programming},
editor = {Turner, David A.},
year = {1990},
isbn = {0-201-17236-4},
pages = {17--42},
numpages = {26},
url = {http://dl.acm.org/citation.cfm?id=119830.119832},
acmid = {119832},
publisher = {Addison-Wesley Longman Publishing Co., Inc.},
address = {Boston, MA, USA},
}
```

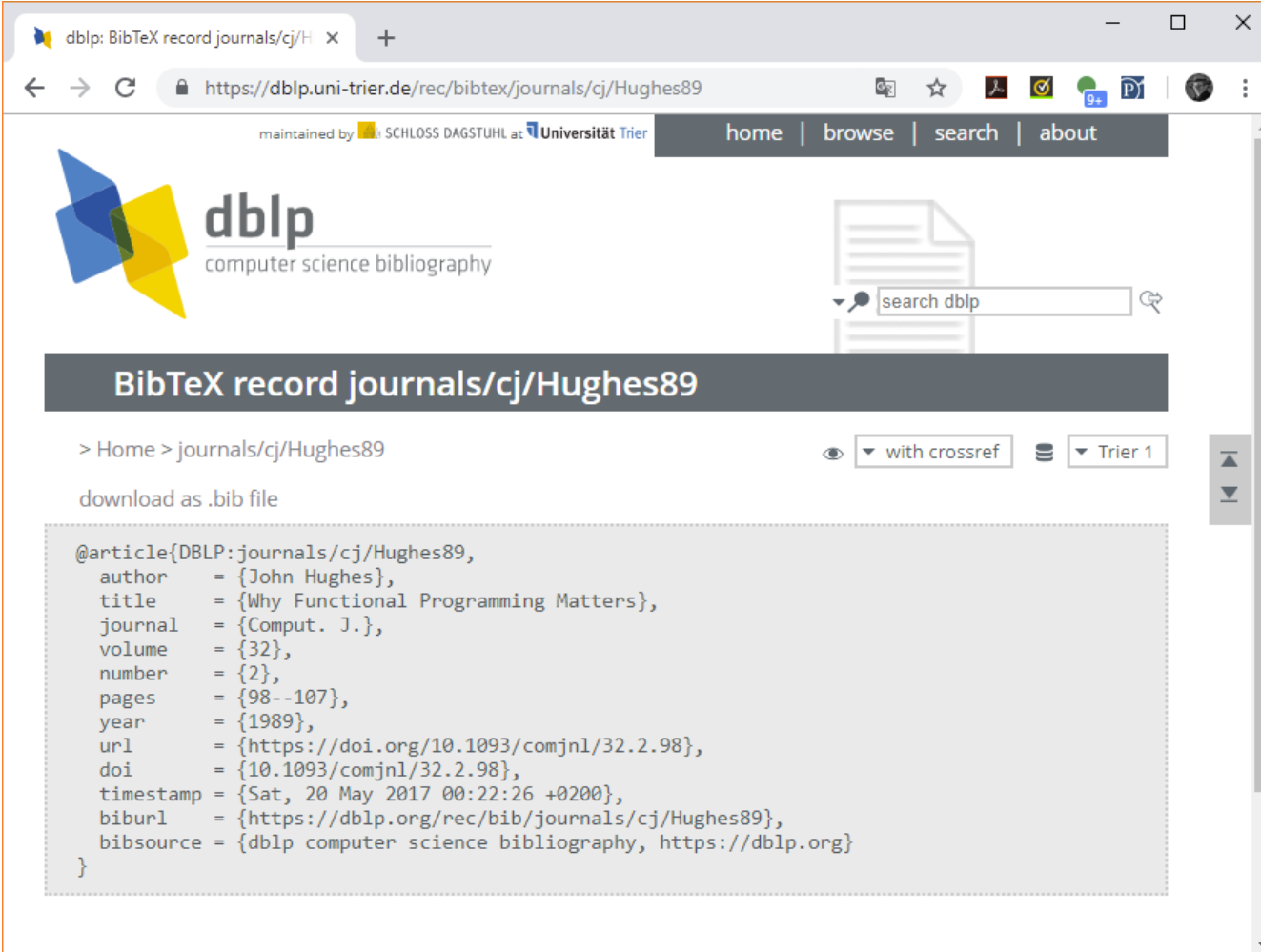
Contact

[download]

<https://dl.acm.org/citation.cfm?id=119832#>

Feedback

# Where do BibTeX entries come from?



The screenshot shows a web browser window displaying the dblp website. The address bar shows the URL `https://dblp.uni-trier.de/rec/bibtex/journals/cj/Hughes89`. The page header includes the dblp logo (computer science bibliography) and navigation links: `home | browse | search | about`. A search bar is visible with the text "search dblp". The main content area displays the title "BibTeX record journals/cj/Hughes89" and a breadcrumb trail: `> Home > journals/cj/Hughes89`. There are two dropdown menus: one for "with crossref" (set to "with crossref") and another for "Trier 1". Below the breadcrumb trail, there is a link to "download as .bib file". The main content is a BibTeX entry enclosed in a dashed box:

```
@article{DBLP:journals/cj/Hughes89,  
  author = {John Hughes},  
  title = {Why Functional Programming Matters},  
  journal = {Comput. J.},  
  volume = {32},  
  number = {2},  
  pages = {98--107},  
  year = {1989},  
  url = {https://doi.org/10.1093/comjnl/32.2.98},  
  doi = {10.1093/comjnl/32.2.98},  
  timestamp = {Sat, 20 May 2017 00:22:26 +0200},  
  biburl = {https://dblp.org/rec/bib/journals/cj/Hughes89},  
  bibsource = {dblp computer science bibliography, https://dblp.org}  
}
```

# LaTeX installations

- MikTeX for Windows
- Sharelatex.com
- ...

Go forth and write...

