HOW TO WRITE PAPERS
SO PEOPLE CAN READ THEM

Derek Dreyer
MPI for Software Systems

PLMW@SPLASH 2019
Athens, Greece
My job as a researcher

Do research
My job as a researcher

- Do research
- Write papers
- Give talks
My job as a researcher

- Do research
- Write papers
- Give talks
Have you read any research papers lately?
Have you read any research papers lately?
Have you read any research papers lately?

- You may think you just lack the technical sophistication to understand them.
Have you read any research papers lately?

- You may think you just lack the technical sophistication to understand them.

- But in fact, many papers are poorly written.
So if you can write clear, accessible papers…

- People will **enjoy** reading them!
- People will **learn** something from them!
- They will get **accepted** to top conferences!
So if you can write clear, accessible papers...

- People will **enjoy** reading them!
- People will **learn** something from them!
- They will get **accepted** to top conferences!

**Fame**
A piece of research
By downcasting the pre-axial gaskets, we achieved 47% reduction in XPS latency on the re-uptake bivalve!
By downcasting the pre-axial gaskets, we achieved 47% reduction in XPS latency on the re-uptake bivalve!

OK, but what does it do, and why do I care?
The good news

• There are **principles** you can follow that will help you write clearer, more readable prose
  
  - Based on how readers process information
The good news

● There are principles you can follow that will help you write clearer, more readable prose

- Based on how readers process information
The good news

- There are principles you can follow that will help you write clearer, more readable prose
  - Based on how readers process information

  “Be clear”
  “Omit needless words”
  ...

- Easy to check if your text satisfies these principles
- If not, principles suggest improvements
The good news

- There are principles you can follow that will help you write clearer, more readable prose
  - Based on how readers process information

    “Be clear”
    “Omit needless words”
    …
The good news

- There are **principles** you can follow that will help you write clearer, more readable prose
  - Based on how readers process information

- These principles are **constructive**:
  - Easy to check if your text satisfies these principles
  - If not, principles suggest improvements
Inspirations for this talk

- **Joseph M. Williams.** *Style: Toward clarity and grace.* 1990. (book)
- **Norman Ramsey.** *Learn technical writing in two hours per week.* (course notes)
- **Simon Peyton Jones.** *How to write a great research paper.* (talk)
Inspirations for this talk

- **Joseph M. Williams.** *Style: Toward clarity and grace.* 1990. (book)

- **Simon Peyton Jones.** *How to write a great research paper.* (talk)

Talk developed jointly with **Rose Hoberman** @ MPI-SWS
Sentences & paragraphs
Flow

It should be clear how each sentence and paragraph relates to the adjacent ones.
Does this text 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.
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.
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.
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.
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.
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.
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.
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.**

What does this game-playing technique have to do with what came before?
Old to new

- Begin sentences with old info
  - Creates link to earlier text

- End sentences with new info
  - Creates link to the text that follows
  - Also places new info in position of emphasis
Applying old-to-new

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.
Applying old-to-new

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.
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.
But flow is not enough!
Lions and tigers are some of the most dramatic and awe-inspiring species of cats. Most of these large cats, however, are currently facing extinction. A smaller cat that has been more evolutionarily successful is the house cat. Although house cats are currently the most popular pet in the world, they are in many ways anti-social. It would therefore be interesting to study whether house cats can be trained to be more sociable.
Lions and tigers are some of the most dramatic and awe-inspiring species of cats. Most of these large cats, however, are currently facing extinction. A smaller cat that has been more evolutionarily successful is the house cat. Although house cats are currently the most popular pet in the world, they are in many ways anti-social. It would therefore be interesting to study whether house cats can be trained to be more sociable.
Lions and tigers are some of the most dramatic and awe-inspiring species of cats. Most of these large cats, however, are currently facing extinction. A smaller cat that has been more evolutionarily successful is the house cat. Although house cats are currently the most popular pet in the world, they are in many ways anti-social. It would therefore be interesting to study whether house cats can be trained to be more sociable.
Lions and tigers are some of the most dramatic and awe-inspiring species of cats. Most of these large cats, however, are currently facing extinction. A smaller cat that has been more evolutionarily successful is the house cat. Although house cats are currently the most popular pet in the world, they are in many ways anti-social. It would therefore be interesting to study whether house cats can be trained to be more sociable.
Lions and tigers are some of the most dramatic and awe-inspiring species of cats. Most of these large cats, however, are currently facing extinction. A smaller cat that has been more evolutionarily successful is the house cat. Although house cats are currently the most popular pet in the world, they are in many ways anti-social. It would therefore be interesting to study whether house cats can be trained to be more sociable.
What about this text?

Lions and tigers are some of the most dramatic and awe-inspiring species of cats. Most of these large cats, however, are currently facing extinction. A smaller cat that has been more evolutionarily successful is the house cat. Although house cats are currently the most popular pet in the world, they are in many ways anti-social. It would therefore be interesting to study whether house cats can be trained to be more sociable.
Lions and tigers are some of the most dramatic and awe-inspiring species of cats. Most of these large cats, however, are currently facing extinction. A smaller cat that has been more evolutionarily successful is the house cat. Although house cats are currently the most popular pet in the world, they are in many ways anti-social. It would therefore be interesting to study whether house cats can be trained to be more sociable.
Lions and tigers are some of the most dramatic and awe-inspiring species of cats. Most of these large cats, however, are currently facing extinction. A smaller cat that has been more evolutionarily successful is the house cat. Although house cats are currently the most popular pet in the world, they are in many ways anti-social. It would therefore be interesting to study whether house cats can be trained to be more sociable.
Coherence

It should be clear how each sentence and paragraph relates to the big picture
One paragraph, one point

• A paragraph should have one main point, expressed in a single point sentence

• Typically the point sentence should appear at or near the beginning of the paragraph
Lions and tigers are some of the most dramatic and awe-inspiring species of cats. Most of these large cats, however, are currently facing extinction. A smaller cat that has been more evolutionarily successful is the house cat. Although house cats are currently the most popular pet in the world, they are in many ways anti-social. It would therefore be interesting to study whether house cats can be trained to be more sociable.
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.
Flow & coherence

Create flow with old to new

Create coherence with one paragraph, one point
Two other principles

- **Name your baby:**
  - Give unique names to things and use them consistently

- **Just in time:**
  - Give information precisely when it is needed, not before
Structure of a research paper
The basic idea

**TOP-DOWN**

Explain your work at multiple levels of abstraction, starting at a high level and getting progressively more detailed.
A structure that works

- Abstract (1-2 paragraphs, 1000 readers)
- Intro (1-2 pages, 100 readers)
- Key ideas (2-3 pages, 50 readers)
- Technical meat (4-6 pages, 5 readers)
- Related work (1-2 pages, 100 readers)
A structure that works

- Abstract (1-2 paragraphs, 1000 readers)
- Intro (1-2 pages, 100 readers)
- Key ideas (2-3 pages, 50 readers)
- Technical meat (4-6 pages, 5 readers)
- Related work (1-2 pages, 100 readers)
The CGI model for an abstract/intro

- **Context:**
  - Set the stage, motivate the general topic

- **Gap:**
  - Explain your specific problem and why existing work does not adequately solve it

- **Innovation:**
  - State what you’ve done that is new, and explain how it helps fill the gap
An abstract for this talk
Learning to write well is an essential part of becoming a successful researcher.
Learning to write well is an essential part of becoming a successful researcher. Unfortunately, many researchers find it very hard to write well because they do not know how to view their text from the perspective of the reader.
Learning to write well is an essential part of becoming a successful researcher. Unfortunately, many researchers find it very hard to write well because they do not know how to view their text from the perspective of the reader. In this talk, we present a simple set of principles for good writing, based on an understanding of how readers process information. Unlike such platitudes as "Be clear" or "Omit needless words", our principles are constructive: one can easily check whether a piece of text satisfies them, and if it does not, the principles suggest concrete ways to improve it.
Introduction

• Like an expanded version of the abstract

• Alternative approach (SPJ): Eliminate Context
  
  – Start with a concrete example, e.g. “Consider this Haskell code…”
  
  – If this works, it can be effective, but I find it often doesn’t work

  – It assumes reader already knows context
A structure that works

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

- Use **concrete illustrative examples** and high-level intuition
- Do **not** have to show the general solution (that's what the technical section is for)
Why have a “key ideas” section at all?

1. Forces you to have a “takeaway”
2. Many readers only care about the takeaway, not the technical details
3. For those who want the technical details, the key ideas are still useful as “scaffolding”
A confession

I don’t always have a key ideas section.
Breadth-first traversal
Breadth-first traversal

Intro:

Key ideas:

Technical meat
Sometimes breadth-first doesn’t work!

e.g., if explaining 3 & 4 requires first explaining subtree rooted at 2
A Promising Semantics for Relaxed-Memory Concurrency

Jeehoon Kang  Chung-Kil Hur∗  Ori Lahav  Viktor Vafeiadis  Derek Dreyer
Seoul National University, Korea  
{jeehoon.kang,gil.hur}@sf.snu.ac.kr
MPI-SWS, Germany †
{orilahav,viktor,dreyer}@mpi-sws.org

Abstract

Despite many years of research, it has proven very difficult to develop a memory model for concurrent programming languages that adequately balances the conflicting desiderata of programmers, compilers, and hardware. In this paper, we propose the first relaxed memory model that (1) accounts for a broad spectrum of features from the C++11 concurrency model, (2) is implementable, in the sense that it provably validates many standard compiler optimizations and reorderings, as well as standard compilation schemes to x86-TSO and Power, (3) justifies simple invariant-based reasoning, thus demonstrating the absence of bad “out-of-thin-air” behaviors, (4) supports “DRF” guarantees, ensuring that programmers who use sufficient synchronization need not understand the full complexities of relaxed-memory semantics, and (5) defines the semantics of racy programs without relying on undefined behaviors, which is a prerequisite for applicability to type-safe languages like Java.

The key novel idea behind our model is the notion of promises: a thread may promise to execute a write in the future, thus enabling other threads to read from that write out of order. Crucially, to memory shared by all threads. To simulate SC semantics on these architectures, one must therefore insert expensive fence instructions to subvert the efforts of the hardware. Secondly, a number of common compiler optimizations—such as constant propagation—are rendered unsound by a naive SC semantics because they effectively reorder memory operations. Moreover, SC semantics is stronger (i.e., more restrictive) than necessary for many concurrent algorithms.

Hence, languages like Java and C++ have opted instead to provide relaxed (aka weak) memory models [22, 13], which enable programmers to demand SC semantics when they need it, but which also support a range of cheaper memory operations that trade off strongly consistent and/or well-defined behavior for efficiency.

1.1 Criteria for a Programming Language Memory Model

Unfortunately, despite many years of research, it has proven very difficult to develop a memory model for concurrent programming languages that adequately balances the conflicting desiderata of programmers, compilers, and hardware. In particular, we would like to find a memory model that satisfies the following properties:
\[
\begin{align*}
\text{(MEMORY: NEW)} & \quad \frac{}{\langle P, M \rangle \xrightarrow{m} \langle P, M \triangleleft m \rangle} \\
\text{(READ-HELPER)} & \quad \frac{o = \text{pln}}{\langle \text{cur.pln}(x) \leq t \rangle} \quad \frac{o \in \{\text{rlx}, \text{ra}\}}{\langle \text{cur.rlx}(x) \leq t \rangle} \\
& \quad \frac{\text{cur}' = \text{cur} \sqcup V \sqcup (o \sqcap \text{ra} \ ? R)}{\langle \text{acq}' = \text{acq} \sqcup V \sqcup (o \sqcap \text{rlx} \ ? R) \rangle} \\
& \quad \text{where } V = [\text{pln} : (o \sqcap \text{rlx} \ ? \{x@t\}), \text{rlx} : \{x@t\}] \\
& \quad \langle \text{cur}, \text{acq}, \text{rel} \rangle \xrightarrow{R.o,x,t,R} \langle \text{cur}', \text{acq}', \text{rel} \rangle \\
\text{(READ)} & \quad \frac{\sigma \xrightarrow{R(o,x,v)}}{\langle \sigma', \text{V}', P \rangle, S, M \rangle} \\
& \quad \frac{\langle x : v \text{w}(, t), R \rangle \in M \quad \mathcal{V} \xrightarrow{R.o,x,t,R}}{\langle \sigma, V, P \rangle, S, M \rangle \rightarrow \langle \sigma', V', P \rangle, S, M \rangle \\
\text{(ACQ-FENCE)} & \quad \frac{\sigma \xrightarrow{F_{\text{acq}}}}{\langle \sigma, \langle \text{cur}, \text{acq}, \text{rel} \rangle, P \rangle, S, M \rangle \rightarrow \langle \sigma', \langle \text{cur}', \text{acq}, \text{rel} \rangle, P \rangle, S, M \rangle} \\
\text{(REL-FENCE)} & \quad \frac{\sigma \xrightarrow{F_{\text{rel}}}}{\langle \sigma', \text{V}', P \rangle, S, M \rangle} \\
\text{(WRITE)} & \quad \frac{\sigma \xrightarrow{\text{pln}(o,x,v)}}{\langle \sigma', \text{V}', P \rangle, S, M \rangle} \\
& \quad \frac{\langle m = \langle x : v \text{w}(, t), R \rangle \rangle \quad \mathcal{V} \xrightarrow{\text{pln}(o,x,t,R)}}{\langle \sigma, V, P \rangle, S, M \rangle \rightarrow \langle \sigma', V', P \rangle, S, M \rangle} \\
\text{(WRITE-HELPER)} & \quad \frac{\text{cur}' = \text{cur} \sqcup V \quad \text{acq}' = \text{acq} \sqcup \text{cur}'}{\langle \text{rel}' = \text{rel}(x) \sqcup V \sqcup (o \sqcap \text{rel} \ ? \text{cur}') \rangle} \\
& \quad \text{where } V = [\text{pln} : \{x@t\}, \text{rlx} : \{x@t\}] \\
& \quad \langle \text{cur}, \text{acq}, \text{rel} \rangle \xrightarrow{W.o,x,t,R,R_w} \langle \text{cur}', \text{acq}', \text{rel}' \rangle \\
\text{(UPDATE)} & \quad \frac{\sigma \xrightarrow{U(o_w,v_w,t,v_w)}}{\langle \sigma', V', P \rangle, S, M \rangle} \\
& \quad \frac{o_w = \text{ra} \quad \forall m' \in P(x). m'.\text{view} = \bot}{\langle \sigma, V, P \rangle, S, M \rangle \rightarrow \langle \sigma', V', P \rangle, S, M \rangle} \\
\text{(SC-FENCE)} & \quad \frac{\sigma \xrightarrow{\text{SystemCall}}}{\langle \sigma', V', P \rangle, S, M \rangle} \\
\text{(MACHINE STEP)} & \quad \frac{\langle \text{TS}(i), S, M \rangle \rightarrow \langle \text{TS}'', S'', M'' \rangle}{\langle \text{TS}'', S'', M'' \rangle \text{ is consistent}} \\
\end{align*}
\]

Figure 3. Full operational semantics.
<table>
<thead>
<tr>
<th>Paragraph</th>
<th>Formula</th>
</tr>
</thead>
</table>
| (MEMORY: NEW)                                                                                                                                                                                             | \[
\langle P, M \rangle \xrightarrow{m} \langle P, M \triangleleft m \rangle
\]                                                                                                                                                                                                 |
| (READ-HELPER)                                                                                                                                                                                             | \[
\begin{align*}
o = \text{pln} & \quad \Rightarrow \quad \text{cur.pln}(x) \leq t \\
o \in \{ \text{rlx}, \text{ra} \} & \quad \Rightarrow \quad \text{cur.rlx}(x) \leq t \\
\text{cur}' = \text{cur} \downarrow \\
\text{acq}' = \text{acq} \downarrow \\
\text{where} \quad V = \{ o = \text{pln} : (o = \text{pln}) \} \\
\langle \text{cur}, \text{acq}, \text{rel} \rangle & \xrightarrow{R} \langle \text{cur}, \text{acq}, \text{rel} \rangle
\end{align*}
\]                                                                                                                                                                                                 |
| (WRITE-HELPER)                                                                                                                                                                                            | \[
\begin{align*}
\text{cur}.\text{rlx}(x) & \leq t \\
\text{cur}' = \text{cur} \sqcup V \\
\text{acq}' = \text{acq} \sqcup \text{cur}'
\end{align*}
\]                                                                                                                                                                                                 |
| (SC-FENCE-HELPER)                                                                                                                                                                                        | \[
\begin{align*}
S' = \text{acq}.\text{rlx} \sqcup S \\
\text{acq}' = \text{acq}' = (S', S') \\
\text{rel}' = \lambda_.\langle S', S' \rangle
\end{align*}
\]                                                                                                                                                                                                 |
| (READ)                                                                                                                                                                                                     | \[
\langle x : v@(-, t], R^x_o, x, t, \rangle \xrightarrow{R(\alpha, x, v)} \langle \sigma, \text{cur}, P', S, M \rangle \xrightarrow{V_{R(\alpha, x, v)}} \langle \sigma', \text{cur}', P', S, M \rangle
\]                                                                                                                                                                                                 |
| (ACQ-FENCE)                                                                                                                                                                                                | \[
\langle \sigma, \text{cur}, \text{acq}, \text{rel} \rangle \xrightarrow{\text{facq}} \langle \sigma', \text{cur}' \rangle
\]                                                                                                                                                                                                 |
| (SILENT)                                                                                                                                                                                                   | \[
\langle \sigma, \text{P}, S, M \rangle \xrightarrow{\text{silent}} \langle \sigma', \text{V}, P \rangle
\]                                                                                                                                                                                                 |
| (PROMISE)                                                                                                                                                                                                  | \[
\begin{align*}
\langle \sigma, \text{P}, S, M \rangle & \xrightarrow{\{ \varepsilon, \hat{c}, \hat{d} \}} \langle \sigma, \text{V}, P \rangle \\
\text{P}' & = \text{P} \xrightarrow{m} \text{M}' \\
\text{M}' & = \text{M} \xrightarrow{m} \text{m}.\text{view} \in \text{M}'
\end{align*}
\]                                                                                                                                                                                                 |
| (MACHINE STEP)                                                                                                                                                                                             | \[
\begin{align*}
\langle \text{TS}(i), \text{S}, \text{M} \rangle & \rightarrow^* \langle \text{TS}', \text{S}', \text{M}' \rangle \\
\langle \text{TS}', \text{S}', \text{M}' \rangle & \xrightarrow{\text{TS}''} \langle \text{TS}''', \text{S}'', \text{M}''' \rangle \\
\langle \text{TS}''', \text{S}'', \text{M}''' \rangle & \text{is consistent}
\end{align*}
\]                                                                                                                                                                                                 |
| **Figure 3.** Full operational semantics.                                                                                                                                                                    | **Figure 3.** Full operational semantics.                                                                                                                                                                   |
Layering the presentation

(THREAD: SILENT)
\[ \sigma \xrightarrow{\text{Silent}} \sigma' \]
\[ \langle \langle \sigma, V, P \rangle, M \rangle \rightarrow \langle \langle \sigma', V, P \rangle, M \rangle \]

(THREAD: READ)
\[ \sigma \xrightarrow{R(x,v)} \sigma' \]
\[ V(x) \leq t \quad V' = V[x \mapsto t] \]
\[ \langle \langle \sigma, V, P \rangle, M \rangle \rightarrow \langle \langle \sigma', V', P \rangle, M \rangle \]

(THREAD: PROMISE)
\[ \langle \langle \sigma, V, P \rangle, M \rangle \rightarrow \langle \langle \sigma, V, P' \rangle, M' \rangle \]

(THREAD: FULFILL)
\[ \sigma \xrightarrow{W(x,v)} \sigma' \]
\[ V(x) < t \quad V' = V[x \mapsto t] \]
\[ \langle \langle \sigma, V', P \rangle, M \rangle \rightarrow \langle \langle \sigma', V', P' \rangle, M \rangle \]

(MACHINE STEP)
\[ \langle \langle \sigma, V, P \rangle, M \rangle \rightarrow \langle \langle \sigma', V', P' \rangle, M' \rangle \]

Figure 1. Operational semantics for the simplified model handling only relaxed read and write accesses.

- **Intro**: A few paragraphs about **main key idea**
- **Section 2**: More details about **main key idea** in a simplified version of the semantics
- **Section 3-4**: Presented **other key ideas** and built up to the full semantics **incrementally**
Layering the presentation

The paper is extremely well written.

The presentation of the semantics is well-motivated and understandable.

- Section 3-4: Presented other key ideas and built up to the full semantics incrementally.
Layering the presentation

- **Intro**: A few paragraphs about **main key idea**

- **Section 2**: More details about **main key idea** in a simplified version of the semantics

- **Section 3-4**: Presented **other key ideas** and built up to the full semantics **incrementally**
Layering the presentation

<table>
<thead>
<tr>
<th>(THREAD: SILENT)</th>
<th>(THREAD: READ)</th>
<th>(THREAD: WRITE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \frac{\sigma \xrightarrow{\text{Silent}} \sigma'}{\langle \sigma, V, P, M \rangle \rightarrow \langle \sigma', V, P, M \rangle} )</td>
<td>( \frac{\sigma \xrightarrow{R(x,v)} \sigma'}{\langle \sigma, V, P, M \rangle \rightarrow \langle \sigma', V, P, M \rangle} )</td>
<td>( \frac{\sigma \xrightarrow{W(x,v)} \sigma'}{\langle \sigma, V, P, M \rangle \rightarrow \langle \sigma', V, P, M \rangle} )</td>
</tr>
<tr>
<td>( \frac{\langle \sigma, V, P', M \rangle \rightarrow \langle \sigma', V, P', M' \rangle} {M' = M \triangleleft m \quad P' = P \triangleleft m} )</td>
<td>( \frac{V(x) \leq t \quad V' = V[x \mapsto t]}{\langle \sigma, V, P, M \rangle \rightarrow \langle \sigma', V', P, M \rangle} )</td>
<td>( \frac{V(x) &lt; t \quad V' = V[x \mapsto t]}{\langle \sigma, V, P, M \rangle \rightarrow \langle \sigma', V', P, M' \rangle} )</td>
</tr>
</tbody>
</table>

Figure 1. Operational semantics for the simplified model handling only relaxed read and write accesses.

- What if you don’t have enough space for such a layered presentation?
  - Move some technical details to appendix
  - Submit to a better conference (i.e. a conference with a higher page limit)
A structure that works

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

1. **It goes at the end** of the paper.
   - You can only properly compare to related work once you’ve explained your own.

2. **Give real comparisons**, not a “laundry list”!
   - Explain in detail how your work fills the **Gap** in a way that related work doesn’t.
Summary of principles

- Flow via “old to new”
- Coherence via “one paragraph, one point”
- Name your baby
- Just in time
- CGI model for abstract/intro
- Layer presentation with “key ideas” section
- Compare with related work at the end