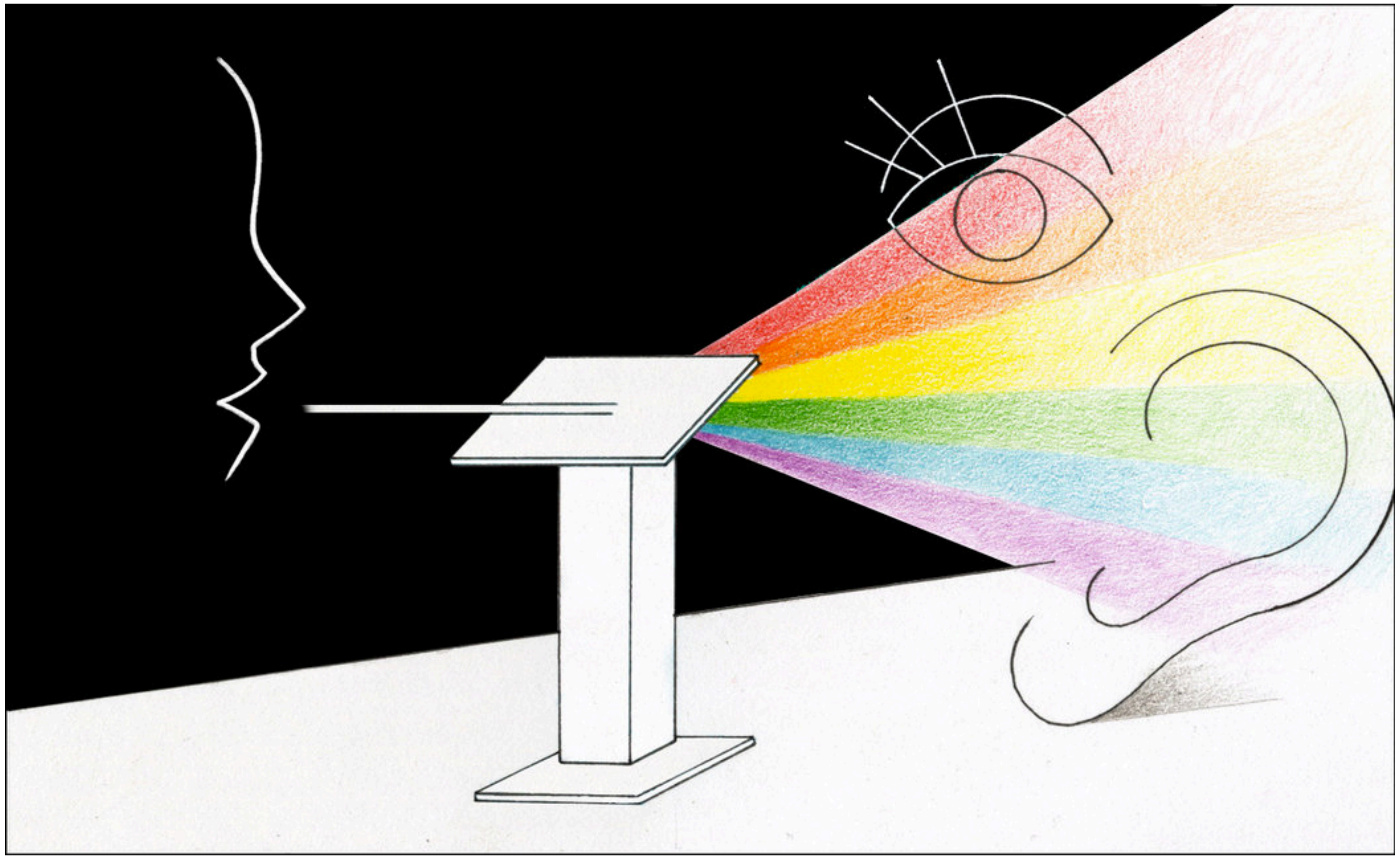
How to Give Talks That People Can Follow



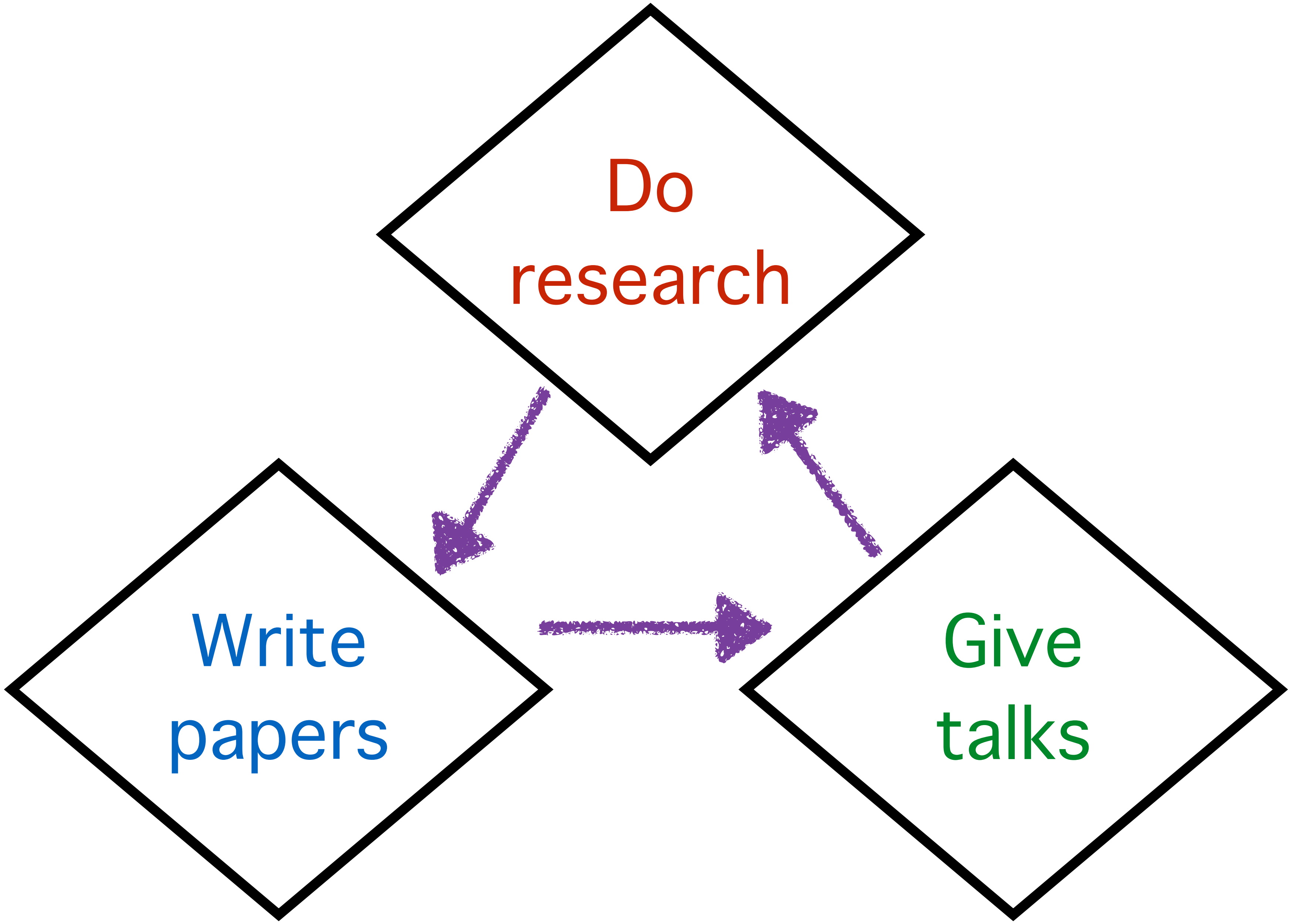
Derek Dreyer

Max Planck Institute for Software Systems

PLMW@PLDI 2021

My job as a researcher

My job as a researcher

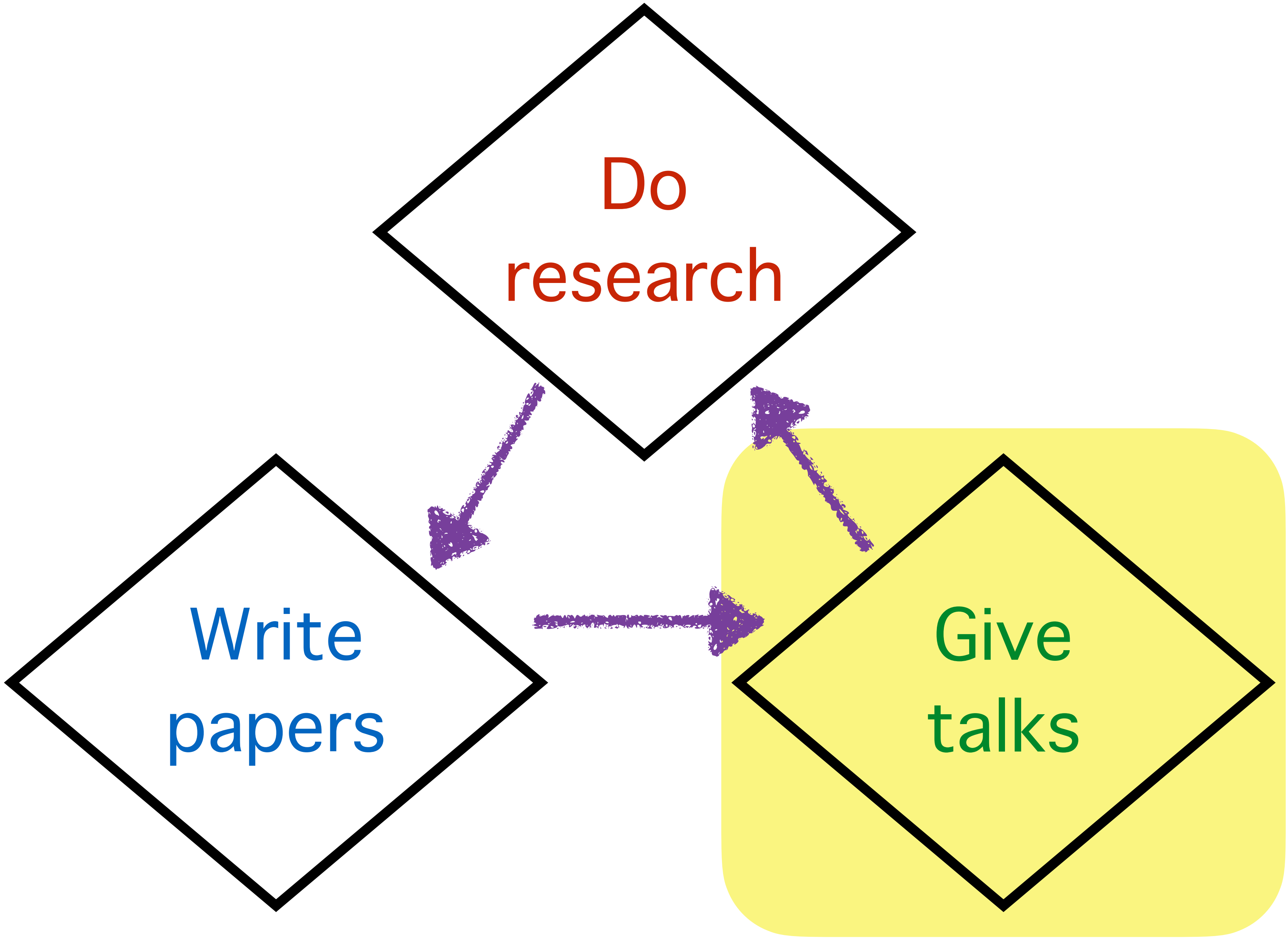


Do research

Write papers

Give talks

My job as a researcher



Do research

Write papers

Give talks

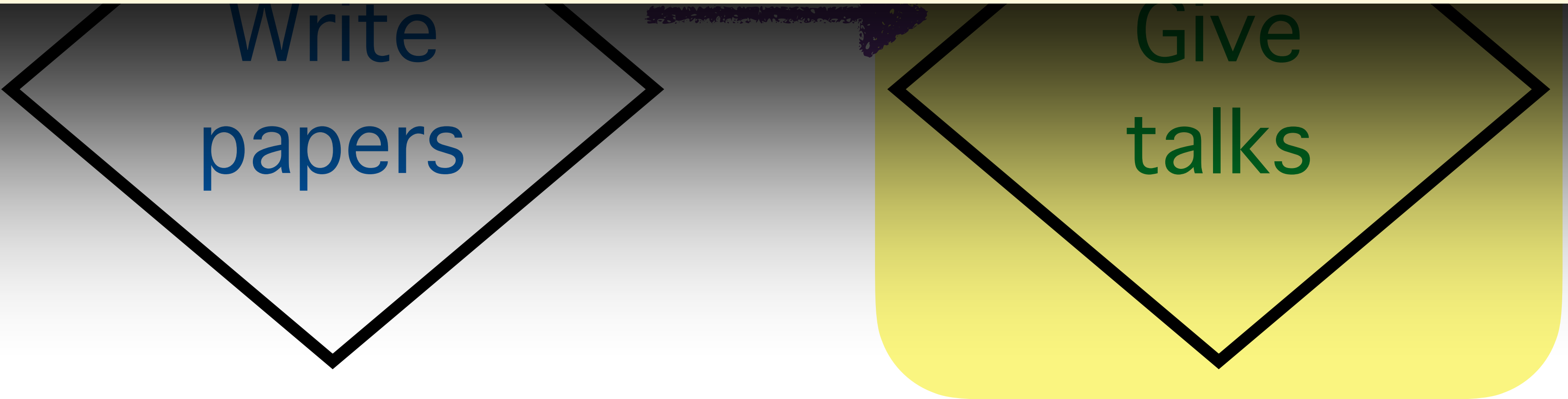
My job as a researcher

Do

Talk developed jointly with

Rose Hoberman

@ MPI-SWS



MIZZOU

Entertain your audience!

- **Simon Peyton Jones.** *How to give a great research talk*. (MSR Summer School, 2016)
  - "Your mission is to **wake them up**!"
  - "Your most potent weapon, by far, is **your enthusiasm**!"

- **John Hughes.** *Unaccustomed as I am to public speaking*. (PLMW, 2016)
  - "**Put on a show**!"

Entertain your audience!

- **Simon Peyton Jones.** *How to give a great research talk*. (MSR Summer School, 2016)

Good advice, but I don’t know how to teach people to be entertaining…

- **John Hughes.** *Unaccustomed as I am to public speaking*. (PLMW, 2016)
  - “**Put on a show**!”

What is your main goal in giving a conference talk?

What is your main goal in giving a conference talk?

Get people to read your paper?

What is your main goal in giving a conference talk?

Get people to read your paper?

No! Talk ≠ Paper

What is your main goal in giving a conference talk?

Give people positive feelings about you and your work!

Extreme example:
My ICFP’15 talk

Pilsner:

A Compositionally Verified Compiler for a Higher-Order Imperative Language

Georg Neis, Chung-Kil Hur,
Jan-Oliver Kaiser, Craig McLaughlin,
Derek Dreyer, Viktor Vafeiadis

**MPI-SWS (Germany),
Seoul National University,
University of Glasgow**

ICFP 2015
Vancouver

Our Contributions

Parametric Inter-Language Simulations (PILS):

- New way to define semantics preservation
- Modular, flexible, ***and*** transitive

Pilsner:

- The ***first*** compositionally verified multi-pass compiler for an ML-like language
- Verified using PILS in Coq!

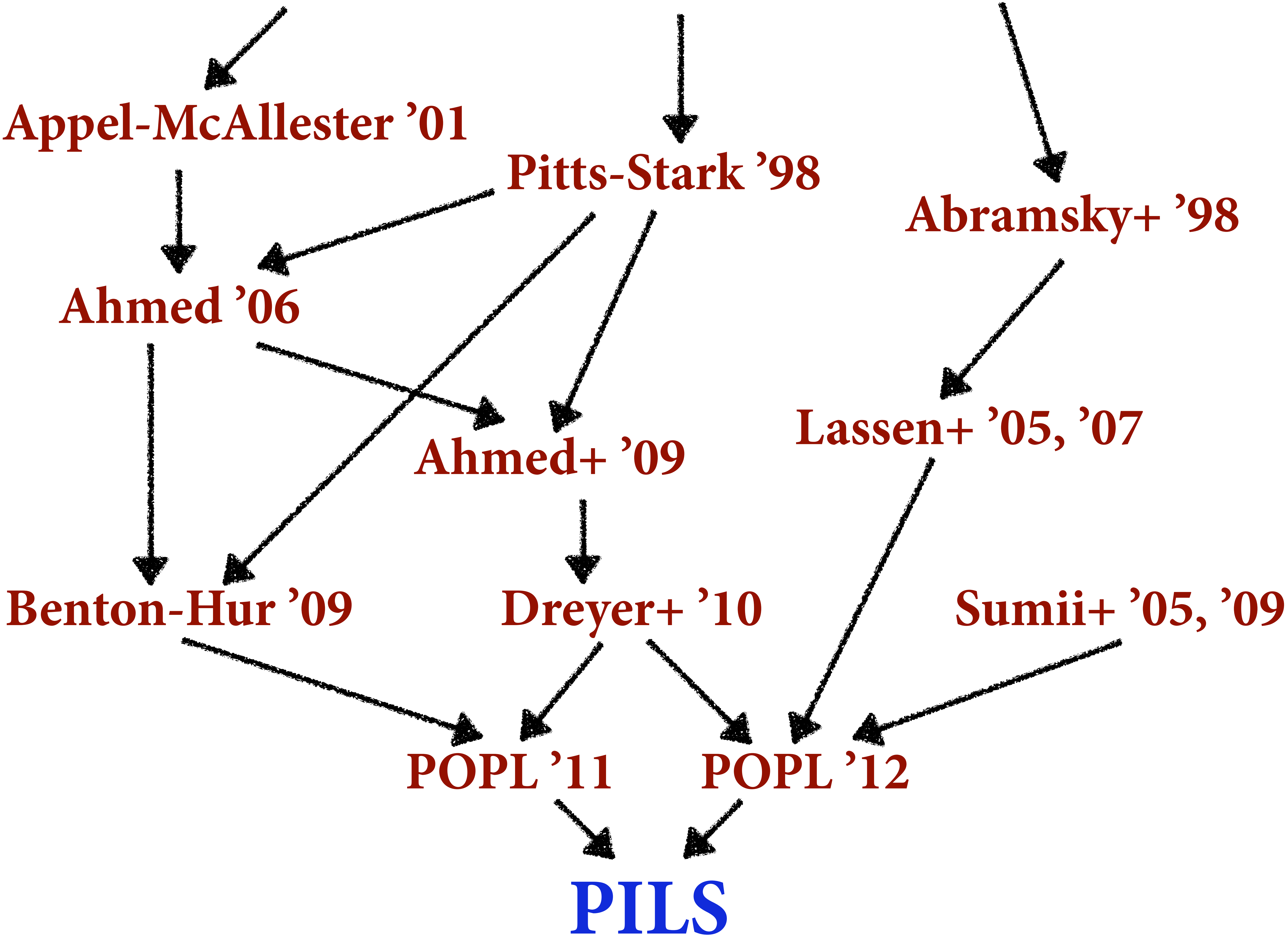
Our Contributions

Parametric Inter-Language Simulations (PILS):

- New way to define semantics preservation
- Modular, flexible, ***and*** transitive

Pilsner:

- The ***first*** compositionally verified multi-pass compiler for an ML-like language
- Verified using PILS in Coq!



Appel-McAllester ’01

Pitts-Stark ’98

Abramsky+ ’98

Ahmed ’06

Lassen+ ’05, ’07

Ahmed+ ’09

Benton-Hur ’09

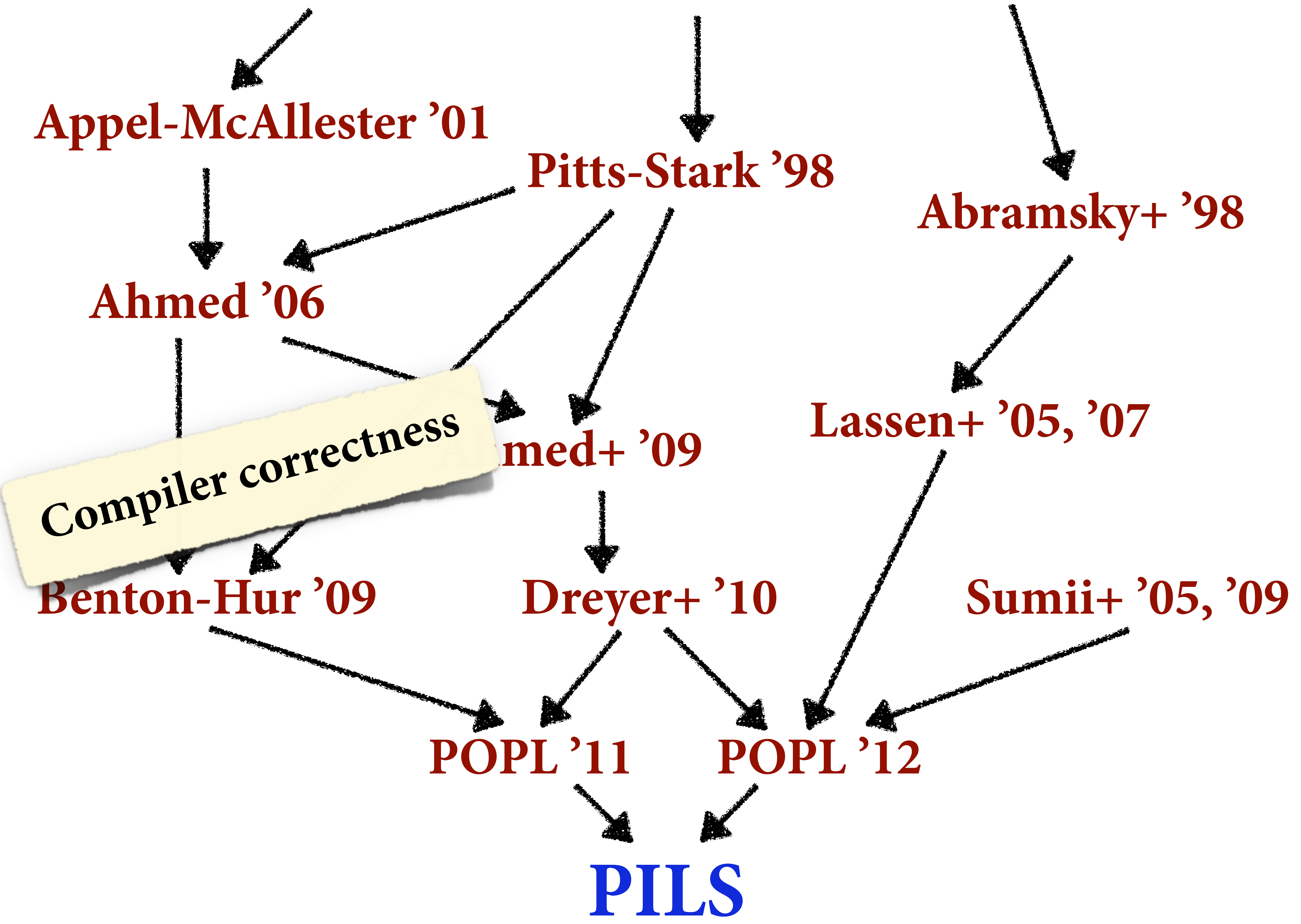
Dreyer+ ’10

Sumii+ ’05, ’09

POPL ’11

POPL ’12

PILS



Appel-McAllester ’01

Pitts-Stark ’98

Abramsky+ ’98

Ahmed ’06

Ahmed+ ’09

Lassen+ ’05, ’07

Compiler correctness

Benton-Hur ’09

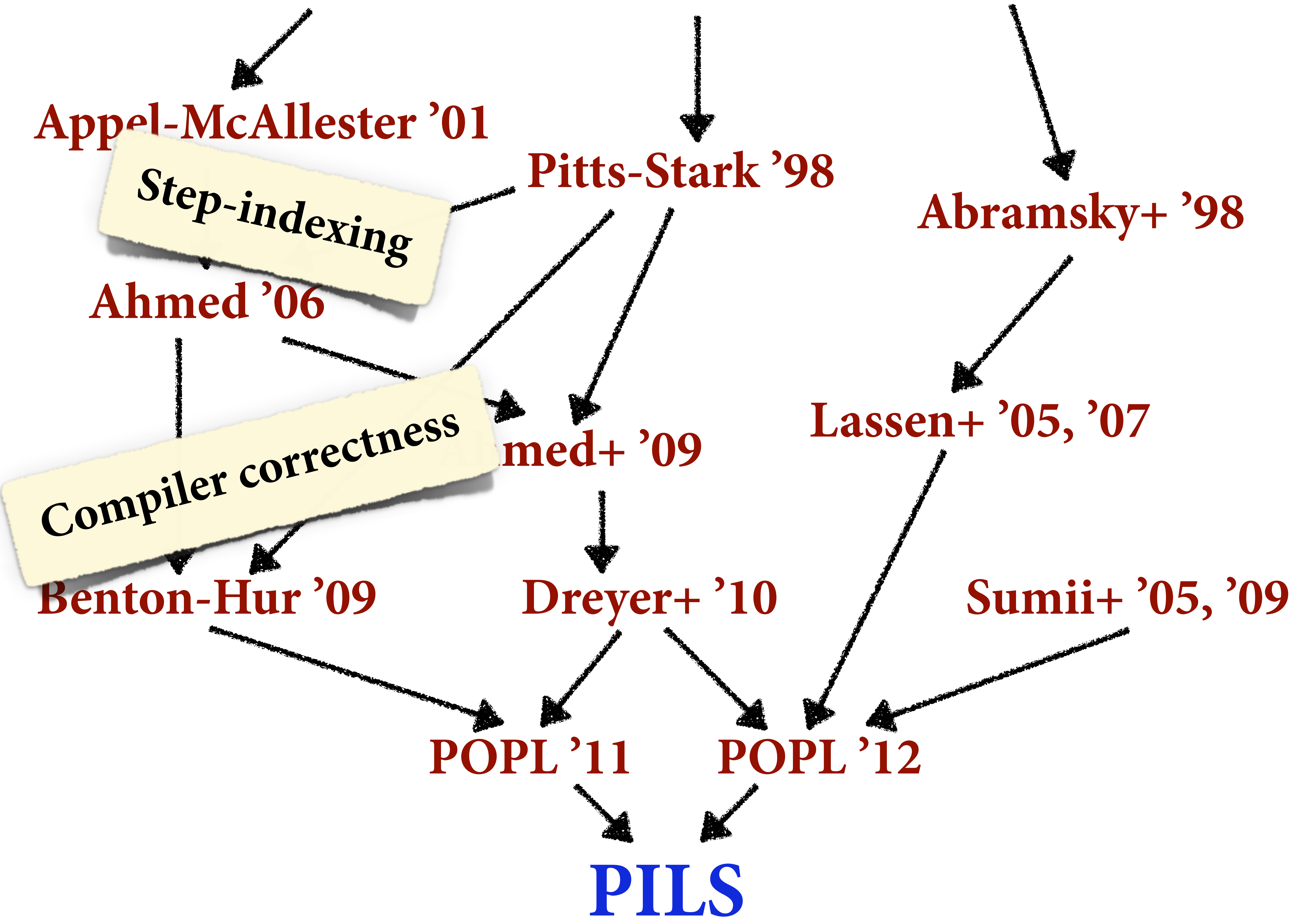
Dreyer+ ’10

Sumii+ ’05, ’09

POPL ’11

POPL ’12

PILS



Appel-McAllester ’01

Pitts-Stark ’98

Abramsky+ ’98

Step-indexing

Ahmed ’06

Ahmed+ ’09

Lassen+ ’05, ’07

Compiler correctness

Benton-Hur ’09

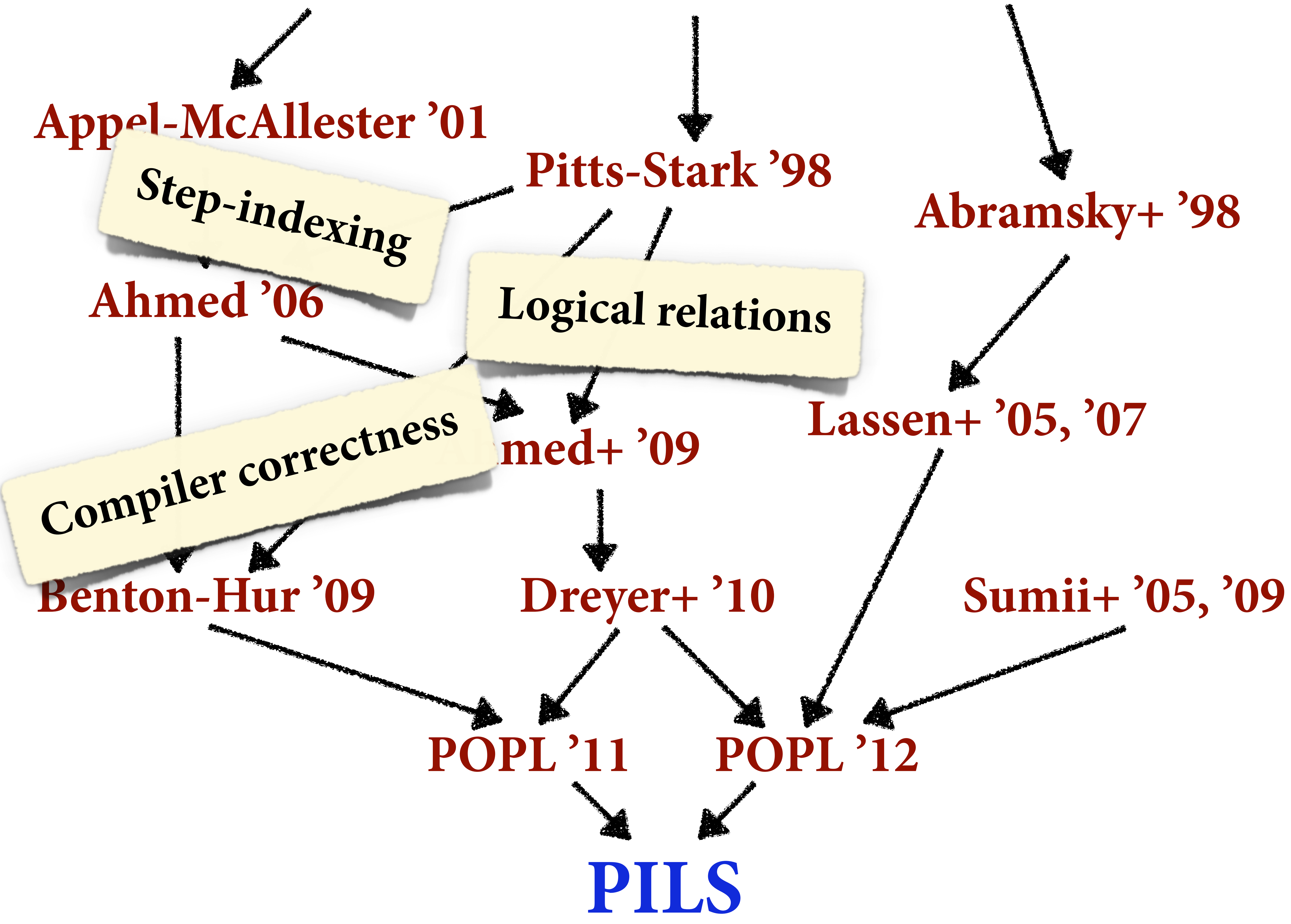
Dreyer+ ’10

Sumii+ ’05, ’09

POPL ’11

POPL ’12

PILS



Appel-McAllester ’01

Pitts-Stark ’98

Abramsky+ ’98

Step-indexing

Ahmed ’06

Logical relations

Lassen+ ’05, ’07

Compiler correctness

Ahmed+ ’09

Benton-Hur ’09

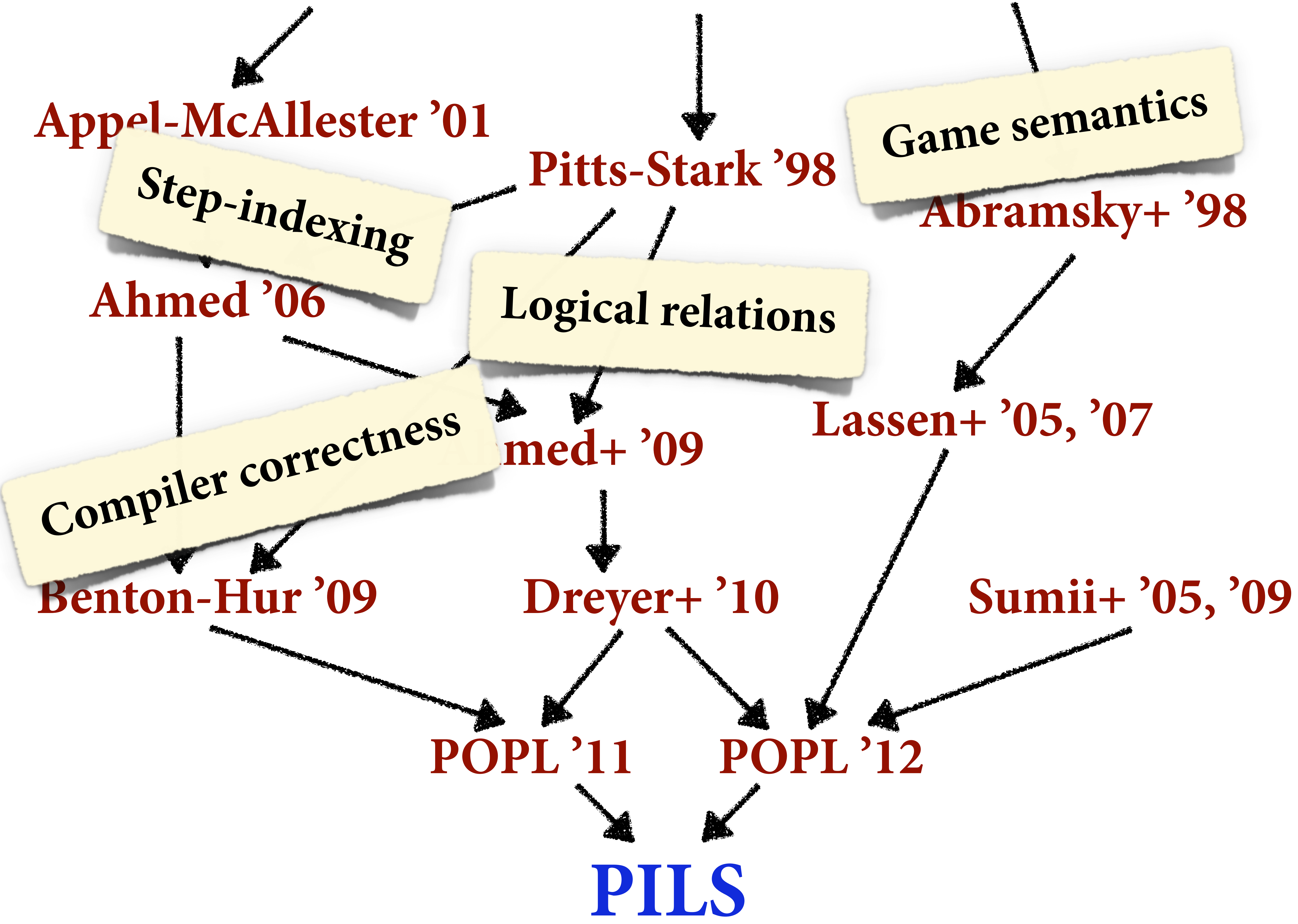
Dreyer+ ’10

Sumii+ ’05, ’09

POPL ’11

POPL ’12

PILS



Appel-McAllester '01

Pitts-Stark '98

Game semantics

Step-indexing

Abramsky+ '98

Ahmed '06

Logical relations

Lassen+ '05, '07

Compiler correctness

Ahmed+ '09

Benton-Hur '09

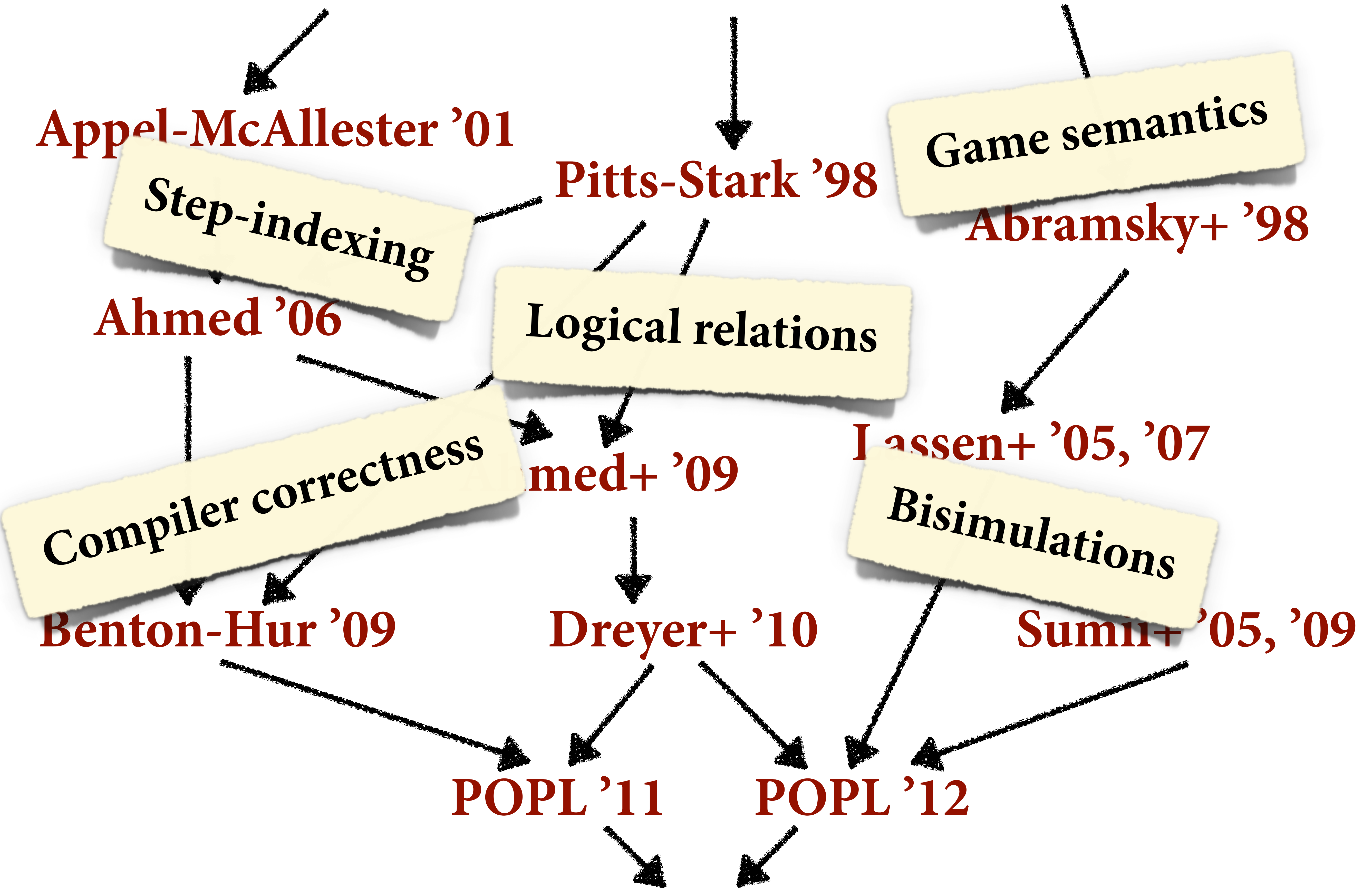
Dreyer+ '10

Sumii+ '05, '09

POPL '11

POPL '12

PILS



Appel-McAllester ’01

Pitts-Stark ’98

Game semantics

Abramsky+ ’98

Step-indexing

Ahmed ’06

Logical relations

Lassen+ ’05, ’07

Ahmed+ ’09

Compiler correctness

Bisimulations

Benton-Hur ’09

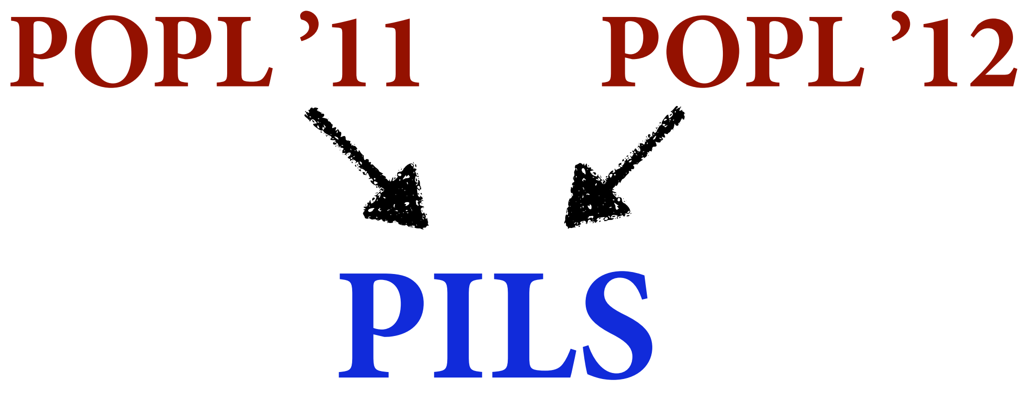
Dreyer+ ’10

Sumii+ ’05, ’09

POPL ’11

POPL ’12

PILS



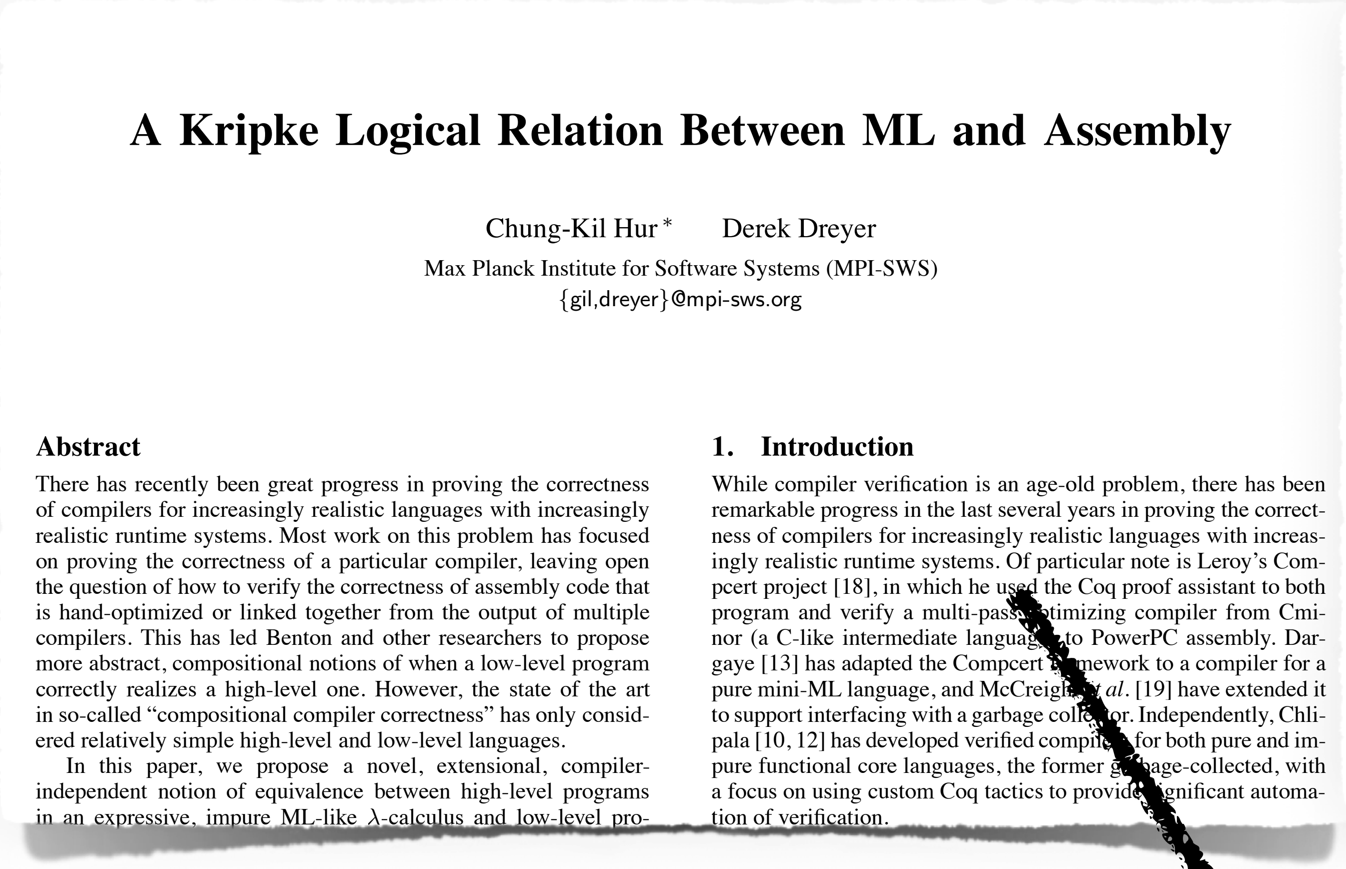
POPL ’11

POPL ’12

PILS

POPL '11

POPL '12



A Kripke Logical Relation Between ML and Assembly

Chung-Kil Hur[*] Derek Dreyer
Max Planck Institute for Software Systems (MPI-SWS)
{gil,dreyer}@mpi-sws.org

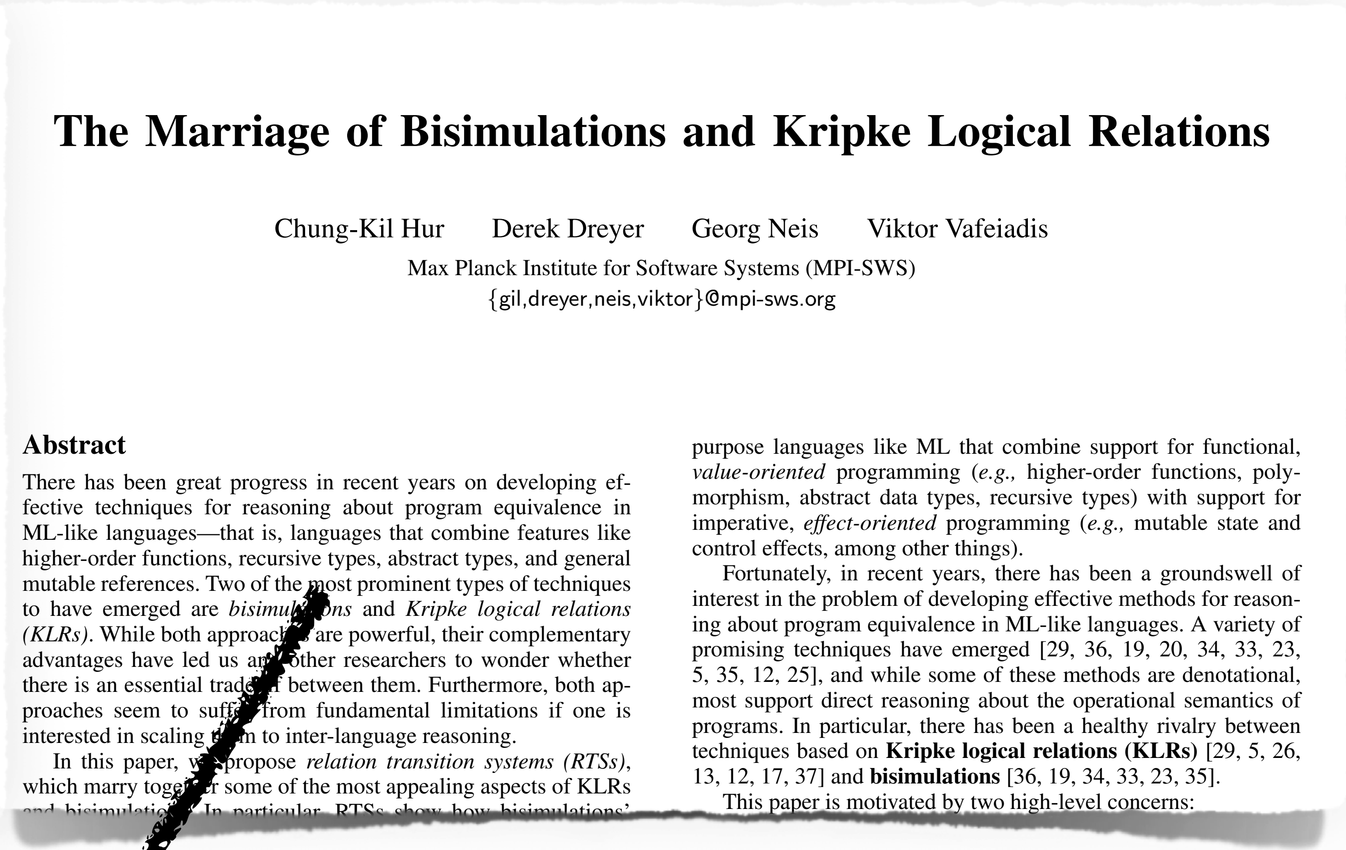
Abstract

There has recently been great progress in proving the correctness of compilers for increasingly realistic languages with increasingly realistic runtime systems. Most work on this problem has focused on proving the correctness of a particular compiler, leaving open the question of how to verify the correctness of assembly code that is hand-optimized or linked together from the output of multiple compilers. This has led Benton and other researchers to propose more abstract, compositional notions of when a low-level program correctly realizes a high-level one. However, the state of the art in so-called "compositional compiler correctness" has only considered relatively simple high-level and low-level languages.

In this paper, we propose a novel, extensional, compiler-independent notion of equivalence between high-level programs in an expressive, impure ML-like λ-calculus and low-level pro-

1. Introduction

While compiler verification is an age-old problem, there has been remarkable progress in the last several years in proving the correctness of compilers for increasingly realistic languages with increasingly realistic runtime systems. Of particular note is Leroy's Compcert project [18], in which he used the Coq proof assistant to both program and verify a multi-pass optimizing compiler from Cminor (a C-like intermediate language) to PowerPC assembly. Dargaye [13] has adapted the Compcert framework to a compiler for a pure mini-ML language, and McCreight *et al.* [19] have extended it to support interfacing with a garbage collector. Independently, Chlipala [10, 12] has developed verified compilers for both pure and impure functional core languages, the former garbage-collected, with a focus on using custom Coq tactics to provide significant automation of verification.



The Marriage of Bisimulations and Kripke Logical Relations

Chung-Kil Hur Derek Dreyer Georg Neis Viktor Vafeiadis
Max Planck Institute for Software Systems (MPI-SWS)
{gil,dreyer,neis,viktor}@mpi-sws.org

Abstract

There has been great progress in recent years on developing effective techniques for reasoning about program equivalence in ML-like languages—that is, languages that combine features like higher-order functions, recursive types, abstract types, and general mutable references. Two of the most prominent types of techniques to have emerged are *bisimulations* and *Kripke logical relations (KLRs)*. While both approaches are powerful, their complementary advantages have led us and other researchers to wonder whether there is an essential tradeoff between them. Furthermore, both approaches seem to suffer from fundamental limitations if one is interested in scaling them to inter-language reasoning.

In this paper, we propose *relation transition systems (RTSs)*, which marry together some of the most appealing aspects of KLRs and bisimulations. In particular, RTSs show how bisimulations'

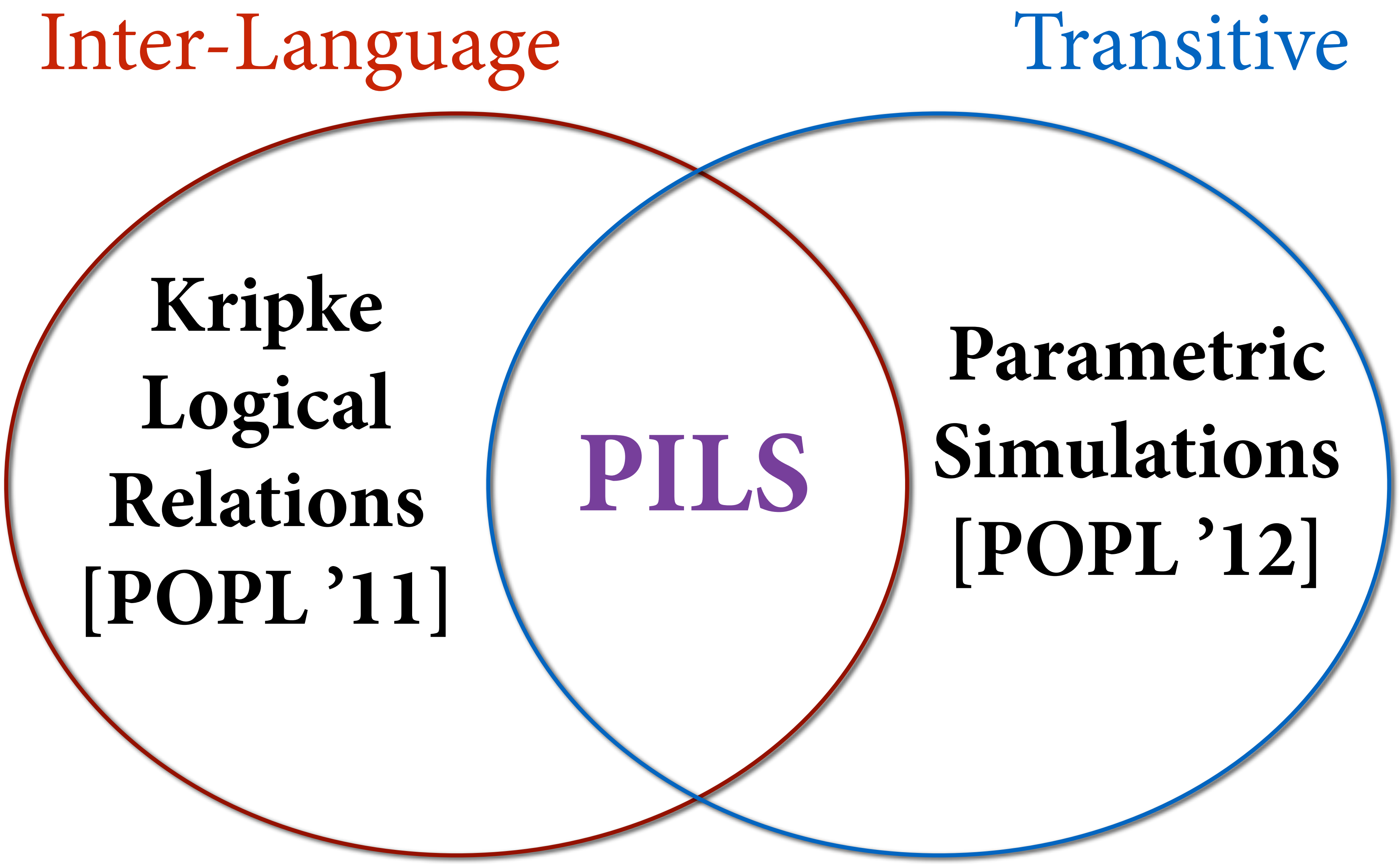
purpose languages like ML that combine support for functional, *value-oriented* programming (*e.g.,* higher-order functions, polymorphism, abstract data types, recursive types) with support for imperative, *effect-oriented* programming (*e.g.,* mutable state and control effects, among other things).

Fortunately, in recent years, there has been a groundswell of interest in the problem of developing effective methods for reasoning about program equivalence in ML-like languages. A variety of promising techniques have emerged [29, 36, 19, 20, 34, 33, 23, 5, 35, 12, 25], and while some of these methods are denotational, most support direct reasoning about the operational semantics of programs. In particular, there has been a healthy rivalry between techniques based on **Kripke logical relations (KLRs)** [29, 5, 26, 13, 12, 17, 37] and **bisimulations** [36, 19, 34, 33, 23, 35].

This paper is motivated by two high-level concerns:

PILS

Putting It Together



Inter-Language

Transitive

Kripke Logical Relations [POPL '11]

PILS

Parametric Simulations [POPL '12]

How is a conference talk different from a paper?

Conference talks

On the plus side:

- ✓ Great advertising for you and your work!

On the minus side:

Conference talks

On the plus side:

✓ Great advertising for you and your work!

On the minus side:

✗ You can’t say much.

✗ The audience may or may not care.

✗ Even those who care will easily get lost.

✗ Slides are a visual medium.

Conference talks

On the plus side:

✓ Great advertising for you and your work!

On the minus side:

✗ You can't say much.

✗ The audience may or may not care.

✗ Even those who care will easily get lost.

✗ Slides are a visual medium.

A paper structure that works

- **Abstract**
- **Intro**
- **Key ideas**
- **Technical meat**
- **Related work**

A paper structure that works

- **Abstract**
- **Intro**
- **Key ideas**
- **Technical meat**
- **Related work**

A ~~paper~~ talk structure that works

- ~~**Abstract**~~
- **Intro**
- **Key ideas**
- ~~**Technical meat**~~
- ~~**Related work**~~

Key ideas

- Use **concrete illustrative examples** and high-level intuition.
- Do **not** show the general solution! (People can go read your paper for that.)

A ~~paper~~ talk structure that works

- ~~Abstract~~
- Intro
- Key ideas
- ~~Technical meat~~
- ~~Related work~~

A ~~paper~~ talk structure that works

- **Intro** (8 minutes)
- **Key ideas** (11 minutes)

A ~~paper~~ talk structure that works

- **Intro** (8 minutes)
- **Key ideas** (11 minutes)
- **What else is in the paper** (1 minute)

Conference talks

On the plus side:

✓ Great advertising for you and your work!

On the minus side:

✗ You can’t say much.

✗ The audience may or may not care.

✗ Even those who care will easily get lost.

✗ Slides are a visual medium.

Conference talks

On the plus side:

✓ Great advertising for you and your work!

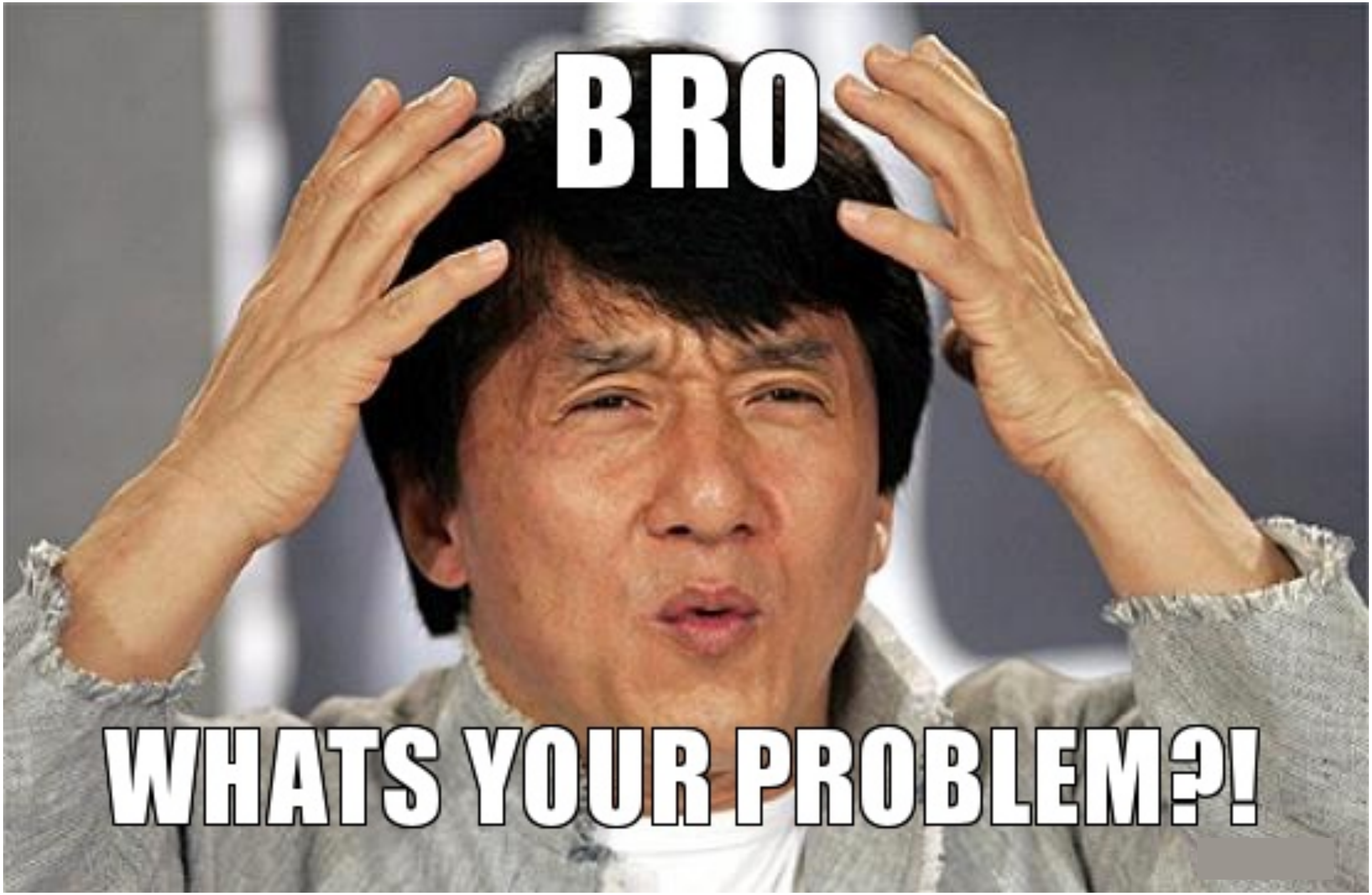
On the minus side:

✗ You can’t say much.

✗ The audience may or may not care.

✗ Even those who care will easily get lost.

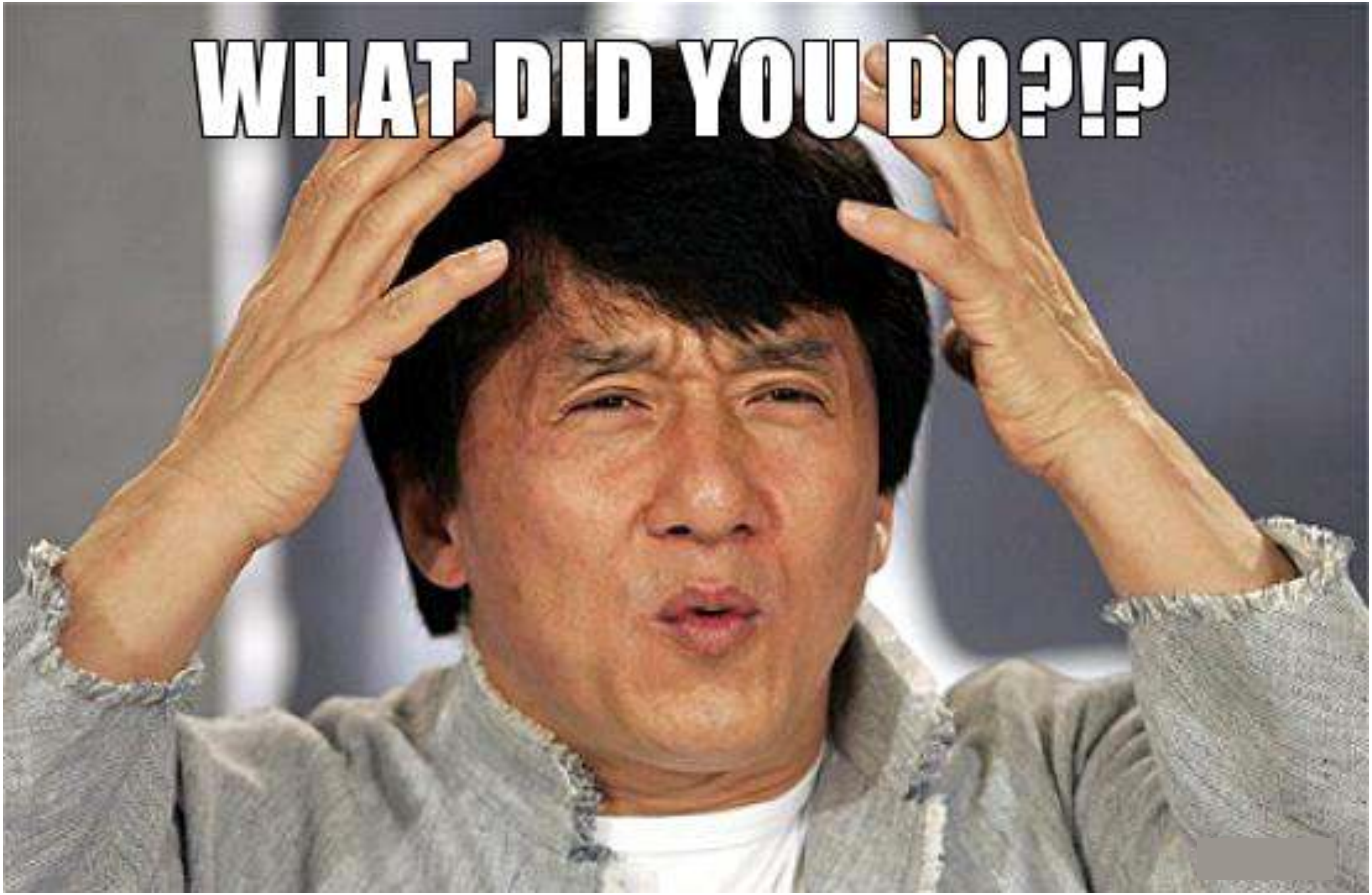
✗ Slides are a visual medium.


BRO
WHATS YOUR PROBLEM?!

Stage the motivation

- **First, get to a problem.**
  - Explain a **general** version of your problem (but not too general) **in the first 2 minutes**.

- **Then, get to the problem.**
  - Motivate and **explicitly state** your **specific** problem in the next 4 minutes.
  - Limit discussion of prior work only to what is needed to explain your problem.


WHAT DID YOU DO?!?

Tell them what you did!

- **Proudly state your contributions.**
  - After the motivation, the audience eagerly wants to hear what you did. Tell them!
- **Follow immediately with a crisp statement of your key idea(s).**
  - It will give audience a take-home message, and give focus to the rest of your talk.

Conference talks

On the plus side:

- ✓ Great advertising for you and your work!

On the minus side:

- ✗ You can’t say much.
- ✗ The audience may or may not care.
- ✗ Even those who care will easily get lost.
- ✗ Slides are a visual medium.

Conference talks

On the plus side:

- ✓ Great advertising for you and your work!

On the minus side:

- ✗ You can't say much.
- ✗ The audience may or may not care.
- ✗ Even those who care will easily get lost.
- ✗ Slides are a visual medium.

A brief diversion into low-level writing skills

Flow

It should be clear how each sentence and paragraph relates to **the adjacent ones**

Does this text flow?

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.

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.

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.

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.

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.

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.

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.

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

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

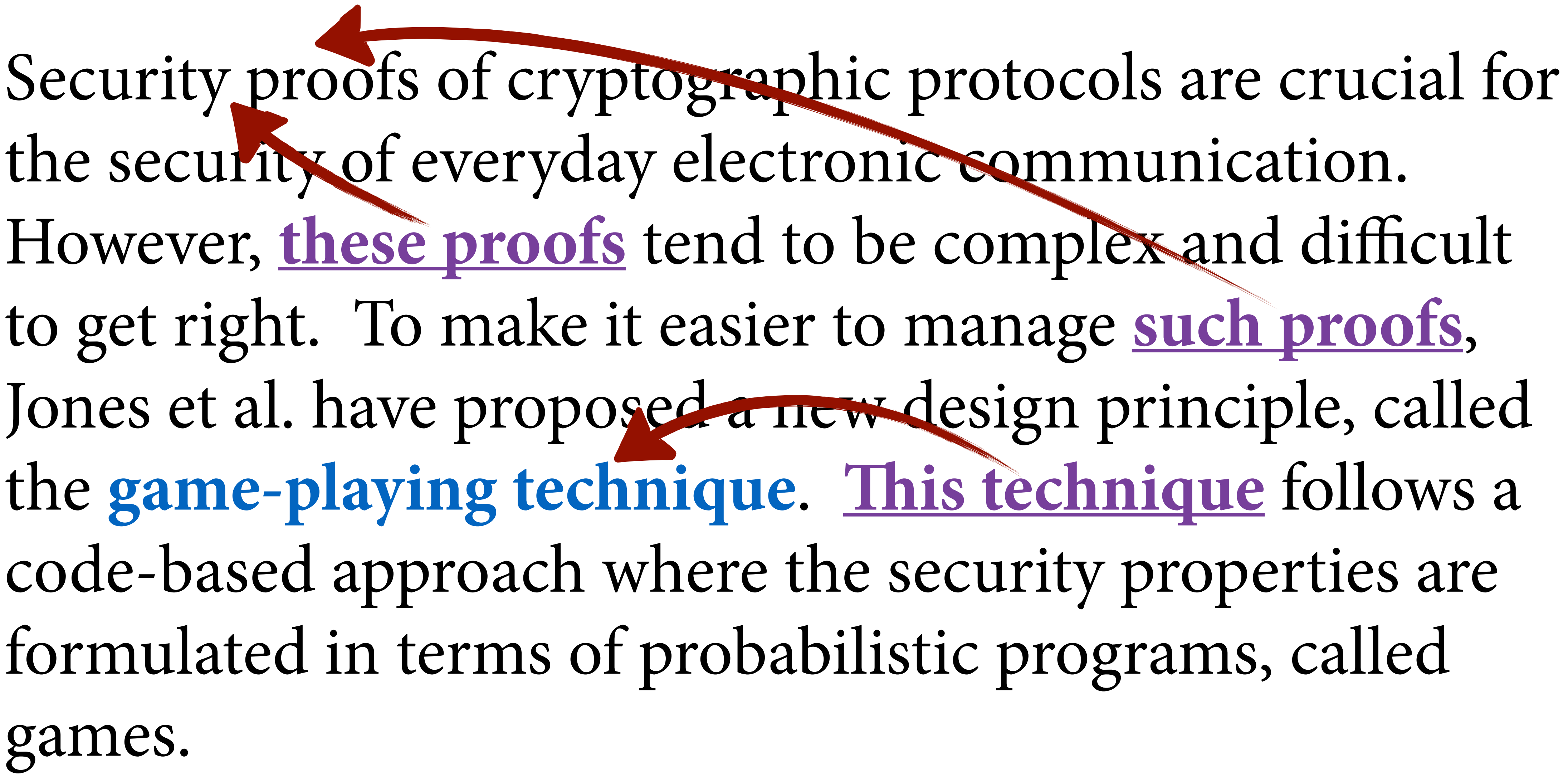
New information

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.

Old-to-new satisfied



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!

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.

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.

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.

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.

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.

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.

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.

What about this text?

Lions and tigers are some of the most matic and
awe-inspiring species of c e large
cats, however n. A
smalle y
successi ats
are curr e world,
they are i ocial. It would
therefore resting to study whether house
cats can be trained to be more sociable.

Has great flow,
but is incoherent!

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**

No point sentence

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.

Point sentence up front

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**

How do flow & coherence apply in giving talks?

Flow in talks

- **Within** a slide:
  - Script should follow “old to new”
- **Between** slides:
  - Don’t just flip to next slide and say, “So…”
  - Plan something to say **during** the transition

Flow & coherence

Create **flow** with **old to new**

Create **coherence** with **one paragraph, one point**

Flow & coherence

Create **flow** with **old to new**

Create **coherence** with

one ~~paragraph~~, one point

slide

Optimization & Concurrency

- Compiler performs several optimizations to generate optimized code.
  - >100 optimizations in GCC, LLVM.

Correct optimizations for sequential programs may be incorrect for shared memory concurrency.

State-of-the-Art:

- Compilers are over-conservative;

 * optimization opportunities are lost.

or

- Buggy optimization

 * *"Premature optimization is the root of all evil"* ~ Donald Knuth

Talklets

- **Break long stretches of talk into talklets.**
  - More digestible units of story (2-4 min.)
  - But just having talklets is not enough…

- **Use transitions between talklets to remind the audience of the big picture.**
  - Summarize the point of the last talklet and how it connects to the next one.

Conference talks

On the plus side:

✓ Great advertising for you and your work!

On the minus side:

✗ You can’t say much.

✗ The audience may or may not care.

✗ Even those who care will easily get lost.

✗ Slides are a visual medium.

Conference talks

On the plus side:

- ✓ Great advertising for you and your work!

On the minus side:

- ✗ You can’t say much.
- ✗ The audience may or may not care.
- ✗ Even those who care will easily get lost.
- ✗ Slides are a visual medium.

Core Idea of Rust

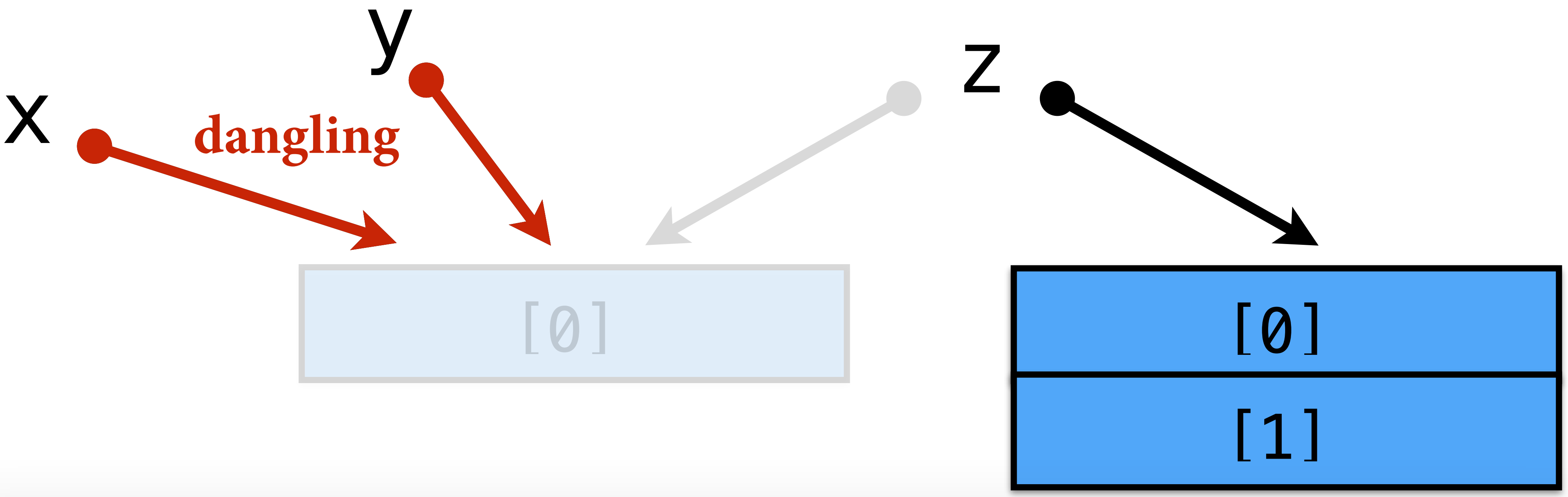
Unrestricted mutation and aliasing lead to:

- use-after-free errors (dangling references)
- data races
- iterator invalidation

~~Mutation~~

~~+~~

~~Aliasing~~

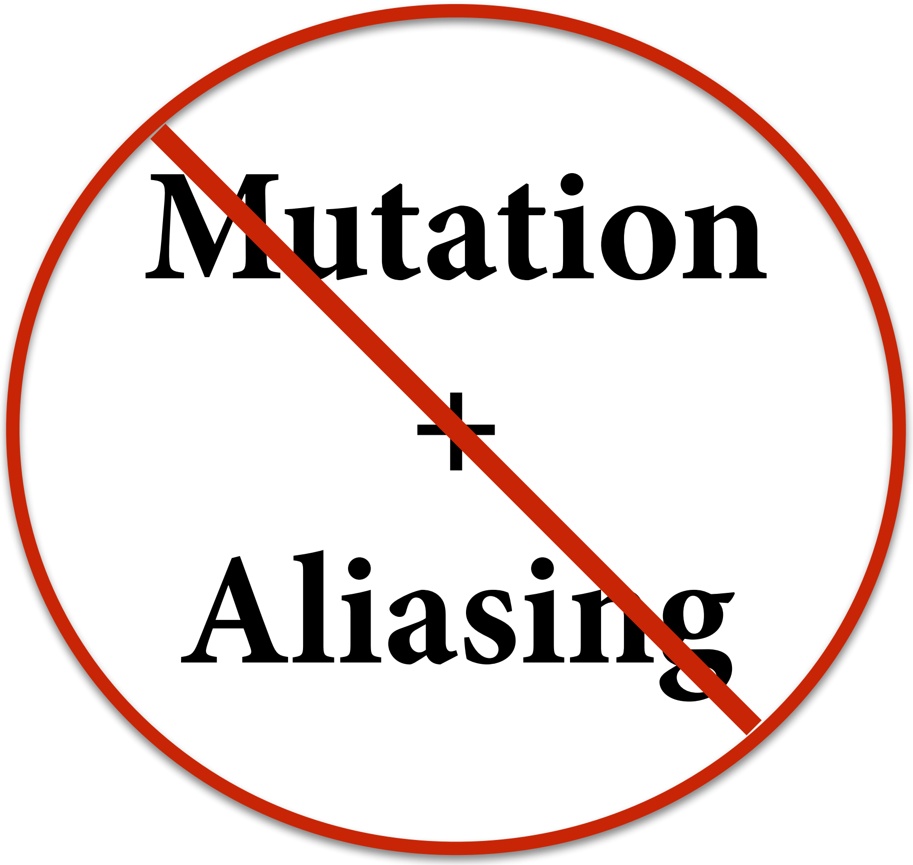


Rust prevents all these errors using a sophisticated “ownership” type system

Make the focus obvious!

(h/t Ranjit Jhala, “How to Design Talks”)

Core Idea of Rust

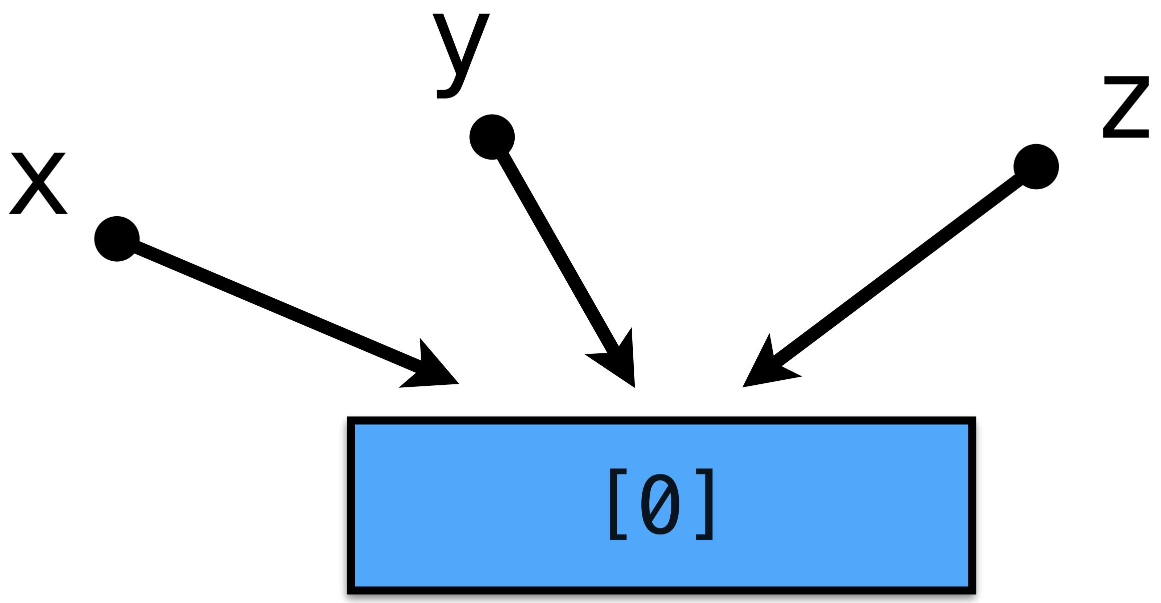


Mutation

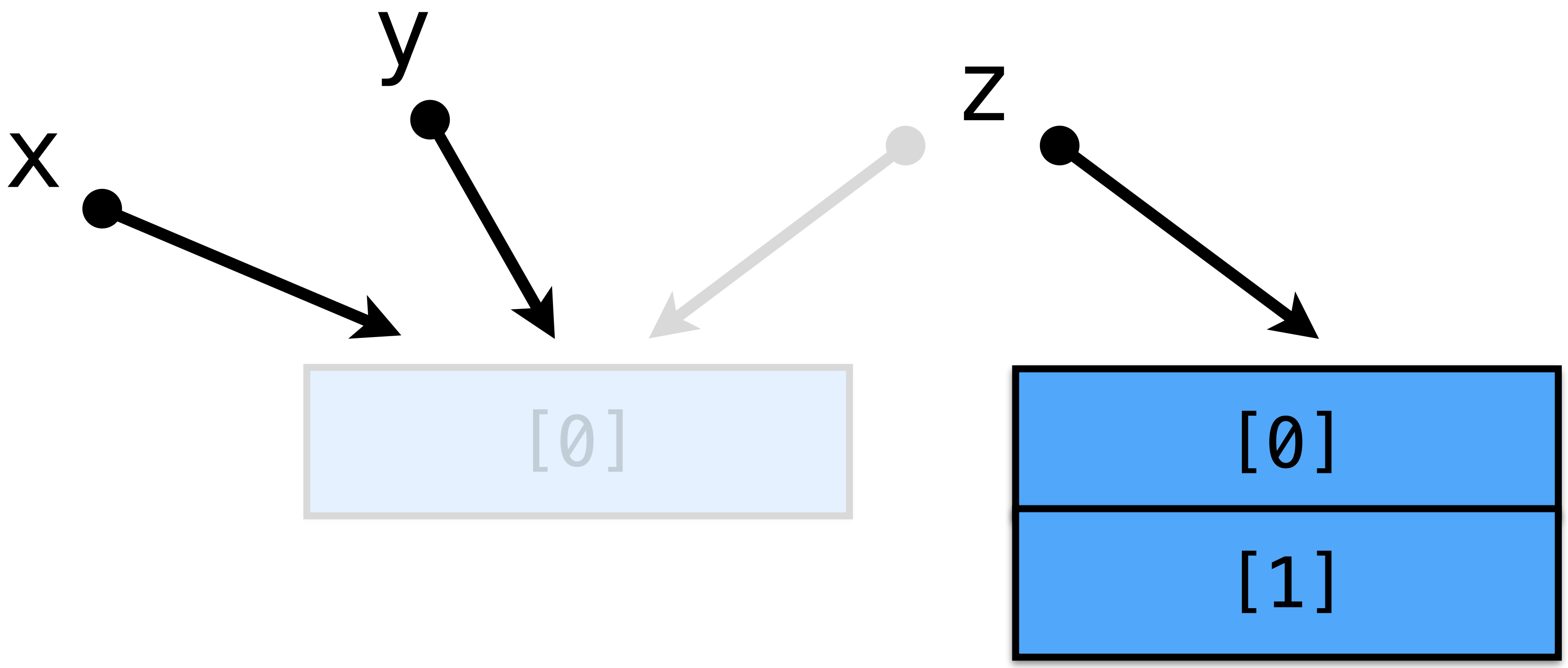
+

Aliasing

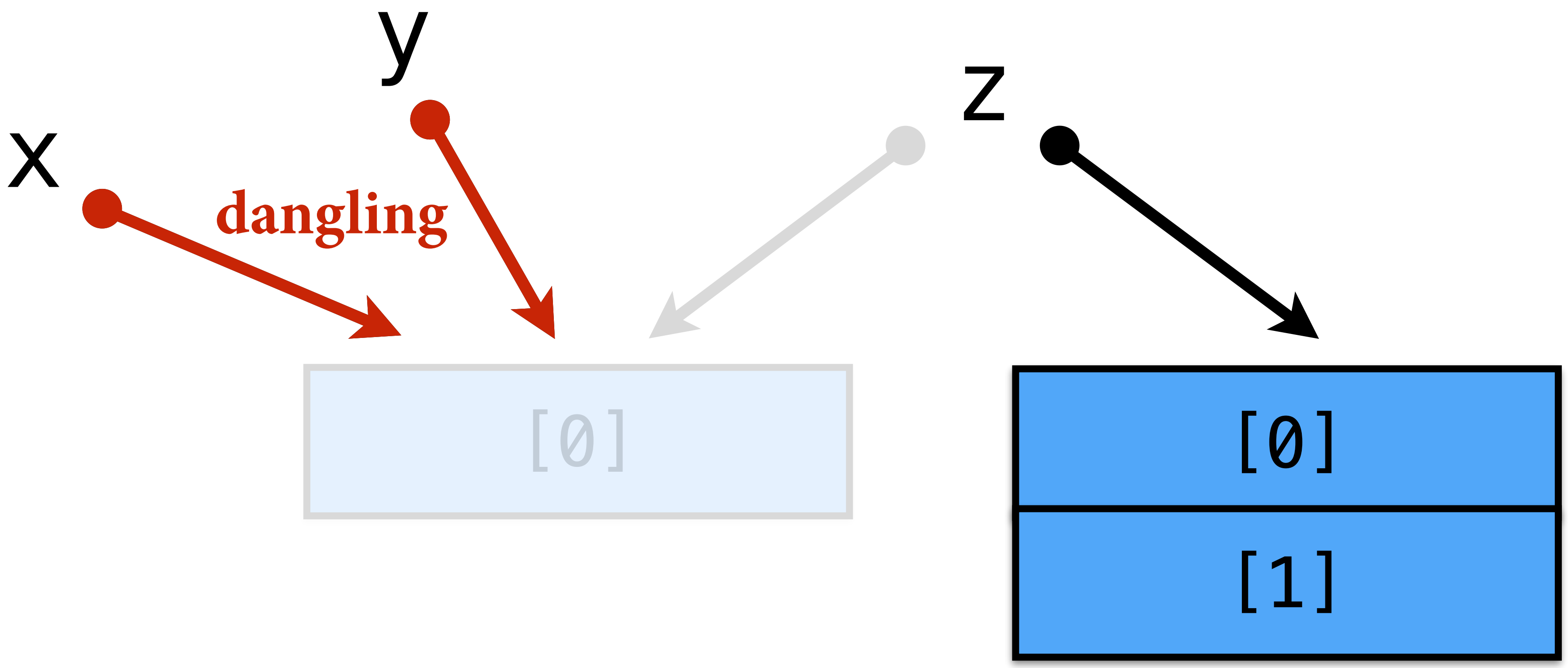
Core Idea of Rust



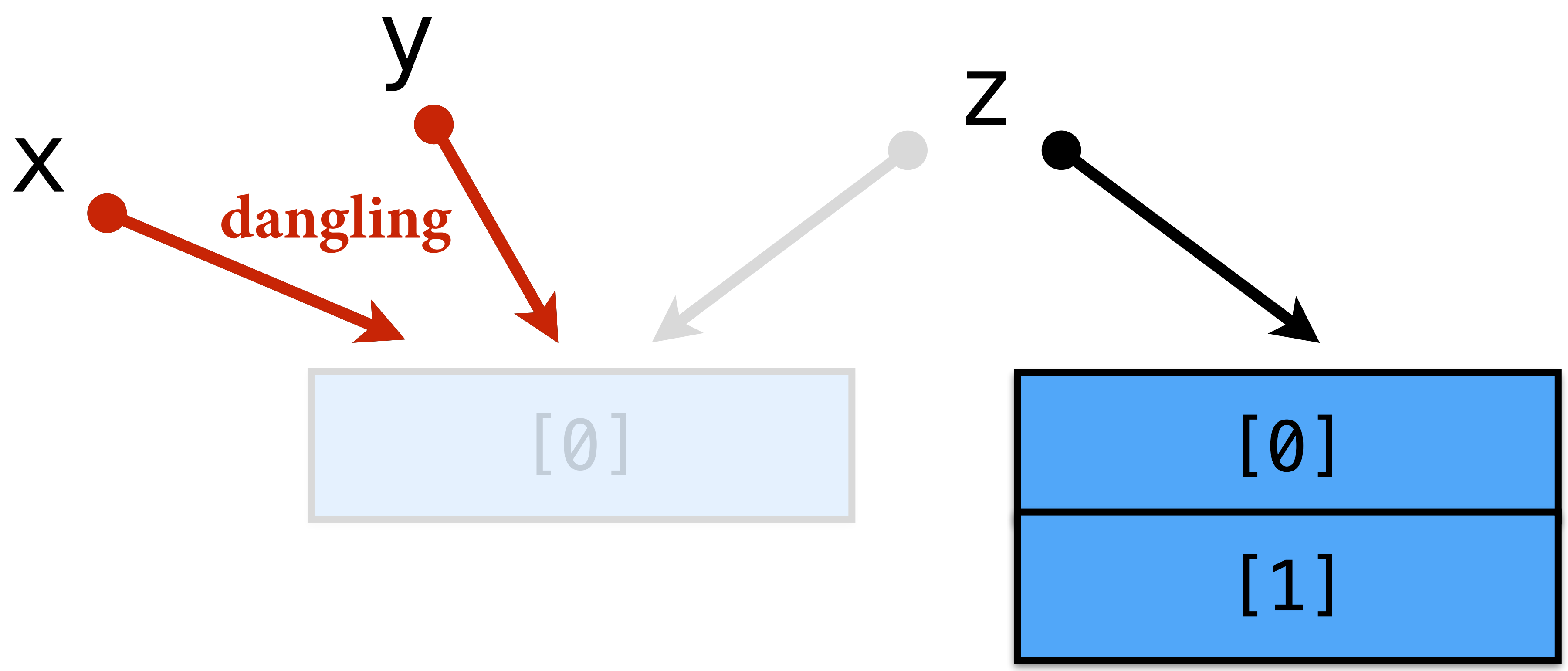
Core Idea of Rust



Core Idea of Rust



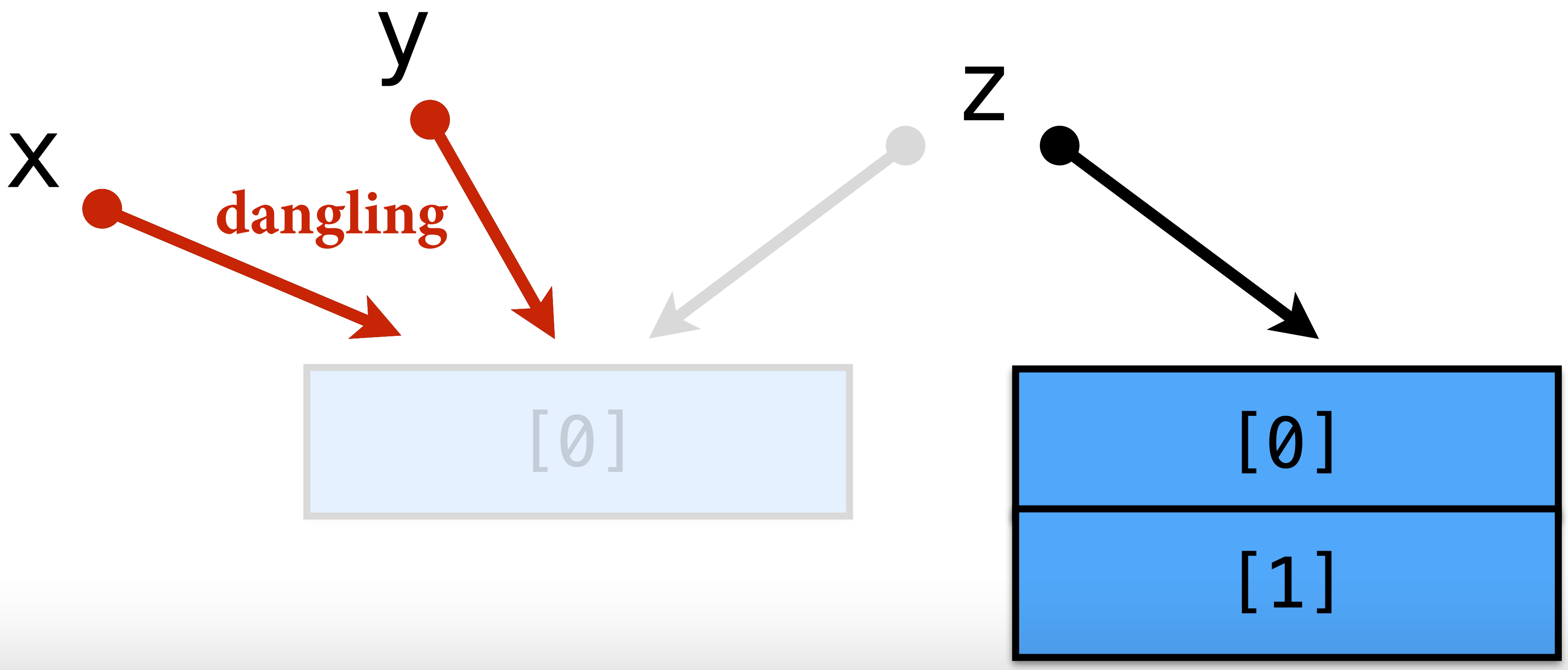
Core Idea of Rust



Unrestricted mutation and aliasing lead to:

- use-after-free errors (dangling references)
- data races
- iterator invalidation

Core Idea of Rust



Rust prevents all these errors using a sophisticated “ownership” type system

One exception to the rule…

Talklets

Talklets

- **Break long stretches of talk into talklets.**

Talklets

- **Break long stretches of talk into talklets.**
  - More digestible units of story (2-4 min.)

Talklets

- **Break long stretches of talk into talklets.**
  - More digestible units of story (2-4 min.)
  - But just having talklets is not enough…

Talklets

- **Break long stretches of talk into talklets.**
  - More digestible units of story (2-4 min.)
  - But just having talklets is not enough…

- **Use transitions between talklets to remind the audience of the big picture.**

Talklets

- **Break long stretches of talk into talklets.**
  - More digestible units of story (2-4 min.)
  - But just having talklets is not enough…
- **Use transitions between talklets to remind the audience of the big picture.**
  - Summarize the point of the last talklet and how it connects to the next one.

Talklets

- **Break long stretches of talk into talklets.**
  - More digestible units of story (2-4 min.)
  - But just having talklets is not enough…
- **Use transitions between talklets to remind the audience of the big picture.**
  - Summarize the point of the last talklet and how it connects to the next one.

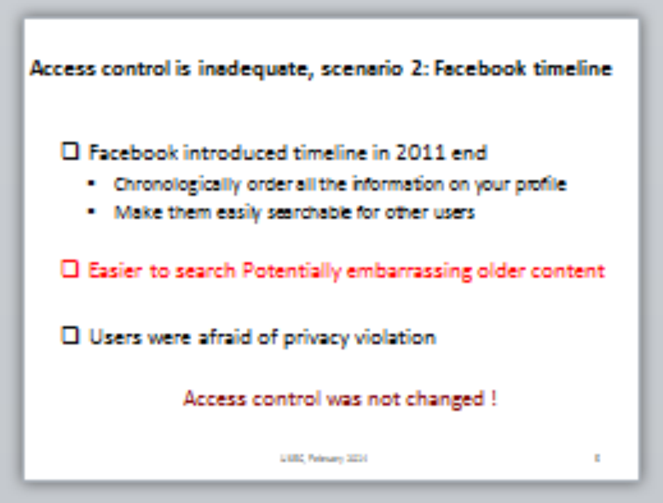
Make the focus obvious

DO:

- Build slide visuals incrementally
- Use smooth animations to clarify transitions

DON’T:

- Reveal bullet points one at a time

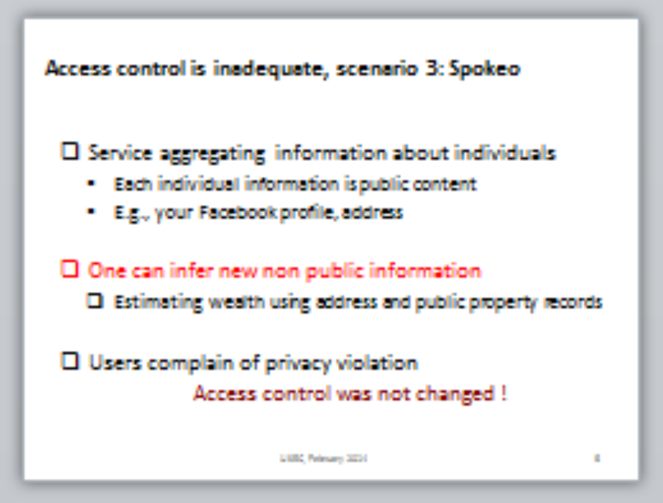


Access control is inadequate, scenario 2: Facebook timeline

- □ Facebook introduced timeline in 2011 end
  - Chronologically order all the information on your profile
  - Make them easily searchable for other users
- □ Easier to search Potentially embarrassing older content
- □ Users were afraid of privacy violation

Access control was not changed !

5

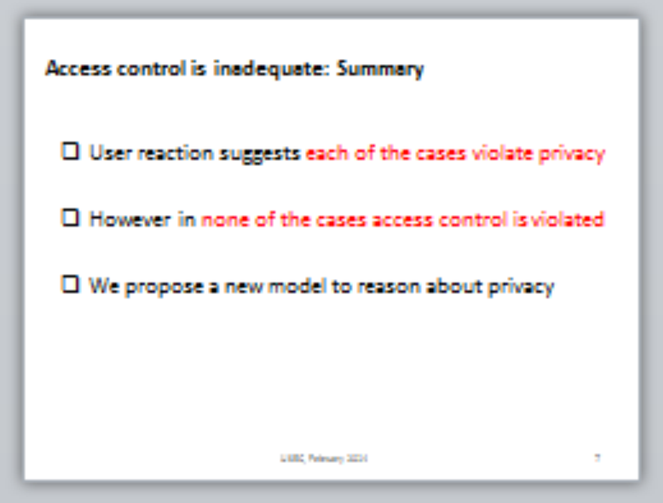


Access control is inadequate, scenario 3: Spokeo

- □ Service aggregating information about individuals
  - Each individual information is public content
  - E.g., your Facebook profile, address
- □ One can infer new non public information
  - □ Estimating wealth using address and public property records
- □ Users complain of privacy violation

Access control was not changed !

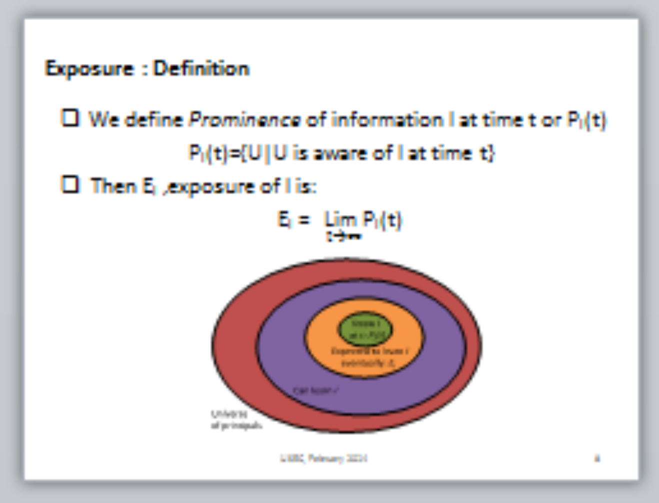
6



Access control is inadequate: Summary

- □ User reaction suggests each of the cases violate privacy
- □ However in none of the cases access control is violated
- □ We propose a new model to reason about privacy

7



Exposure : Definition

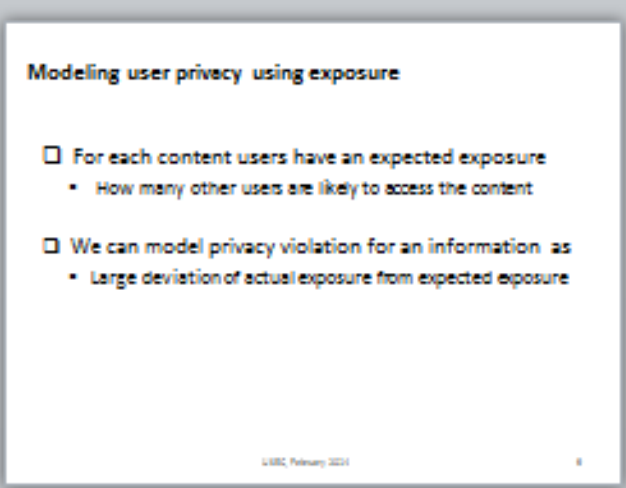
- □ We define *Prominence* of information I at time t or $P_I(t)$

$$P_I(t)=\{U|U \text{ is aware of I at time } t\}$$

- □ Then E_I ,exposure of I is:

$$E_I = \lim_{t\to\infty} P_I(t)$$

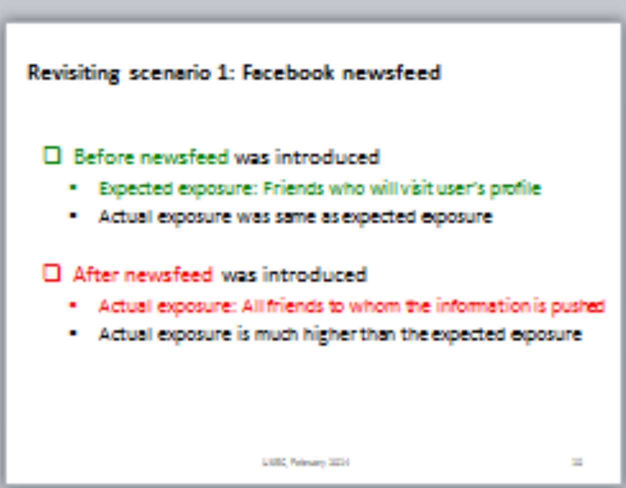
8



Modeling user privacy using exposure

- □ For each content users have an expected exposure
  - How many other users are likely to access the content
- □ We can model privacy violation for an information as
  - Large deviation of actual exposure from expected exposure

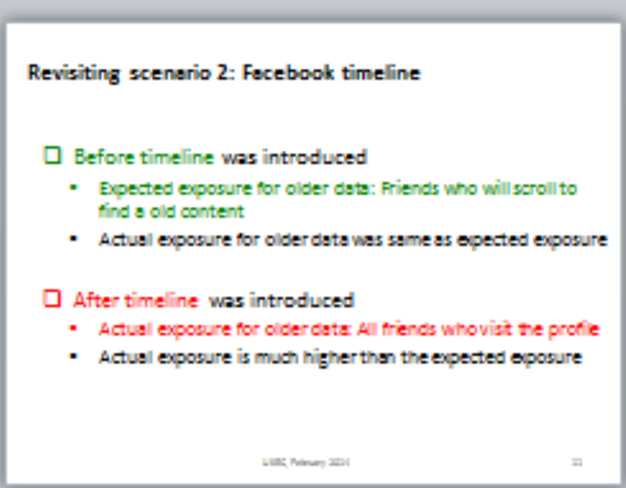
9



Revisiting scenario 1: Facebook newsfeed

- □ Before newsfeed was introduced
  - Expected exposure: Friends who will visit user's profile
  - Actual exposure was same as expected exposure
- □ After newsfeed was introduced
  - Actual exposure: All friends to whom the information is pushed
  - Actual exposure is much higher than the expected exposure

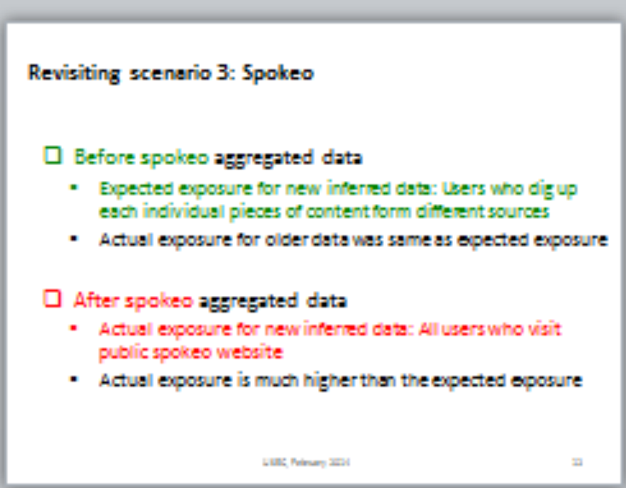
10



Revisiting scenario 2: Facebook timeline

- □ Before timeline was introduced
  - Expected exposure for older data: Friends who will scroll to find a old content
  - Actual exposure for older data was same as expected exposure
- □ After timeline was introduced
  - Actual exposure for older data: All friends who visit the profile
  - Actual exposure is much higher than the expected exposure

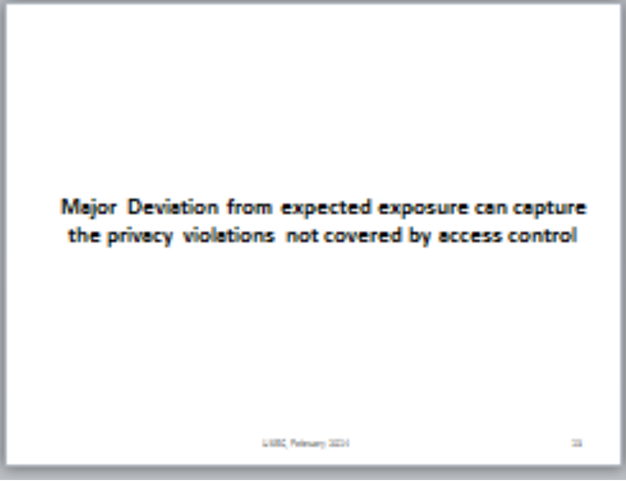
11



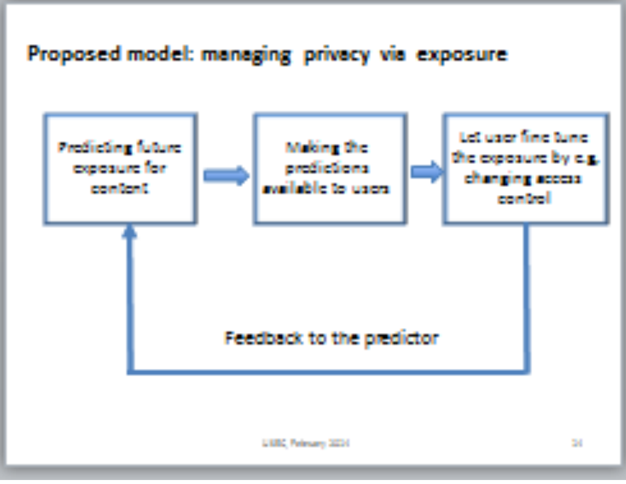
Revisiting scenario 3: Spokeo

- □ Before spokeo aggregated data
  - Expected exposure for new inferred data: Users who dig up each individual pieces of content form different sources
  - Actual exposure for older data was same as expected exposure
- □ After spokeo aggregated data
  - Actual exposure for new inferred data: All users who visit public spokeo website
  - Actual exposure is much higher than the expected exposure

12

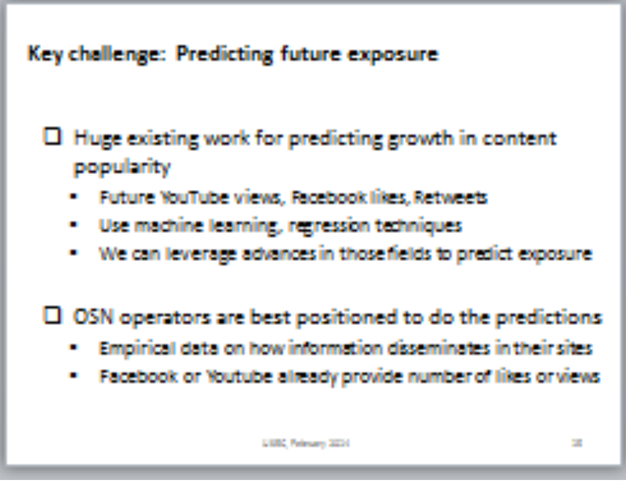


Major Deviation from expected exposure can capture the privacy violations not covered by access control



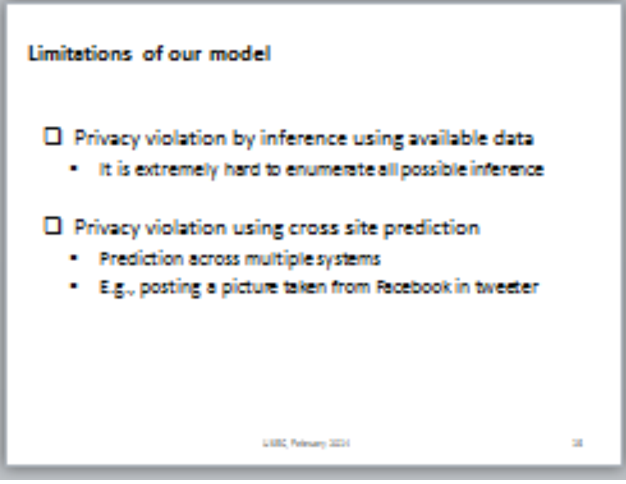
Proposed model: managing privacy via exposure

Predicting future exposure for content
Making the predictions available to users
Let user fine tune the exposure by e.g. changing access control
Feedback to the predictor



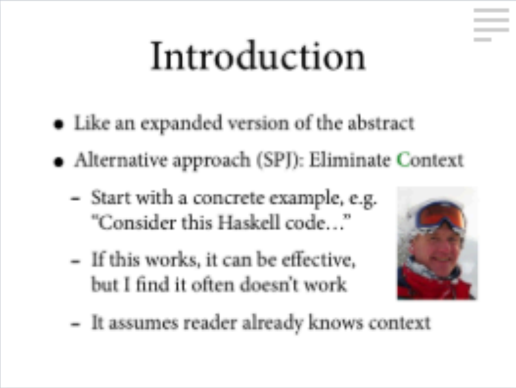
Key challenge: Predicting future exposure

- □ Huge existing work for predicting growth in content popularity
  - Future YouTube views, Facebook likes, Retweets
  - Use machine learning, regression techniques
  - We can leverage advances in those fields to predict exposure
- □ OSN operators are best positioned to do the predictions
  - Empirical data on how information disseminates in their sites
  - Facebook or Youtube already provide number of likes or views



Limitations of our model

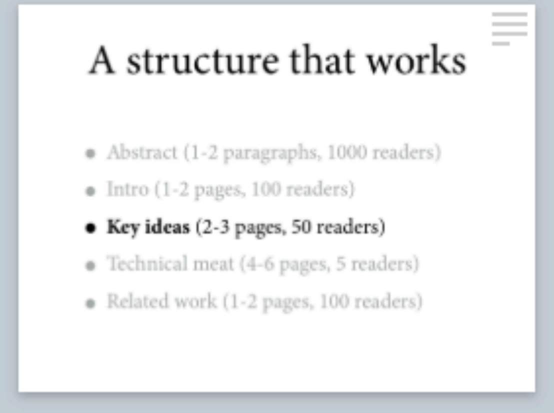
- □ Privacy violation by inference using available data
  - It is extremely hard to enumerate all possible inference
- □ Privacy violation using cross site prediction
  - Prediction across multiple systems
  - E.g., posting a picture taken from Facebook in tweeter



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

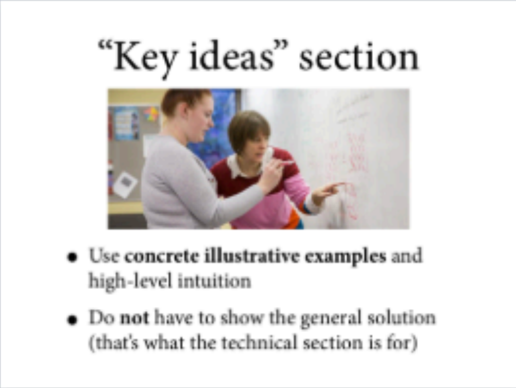
53



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)

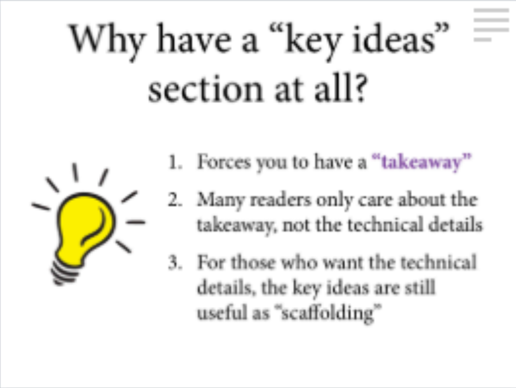
54



"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)

55



Why have a "key ideas" section at all?

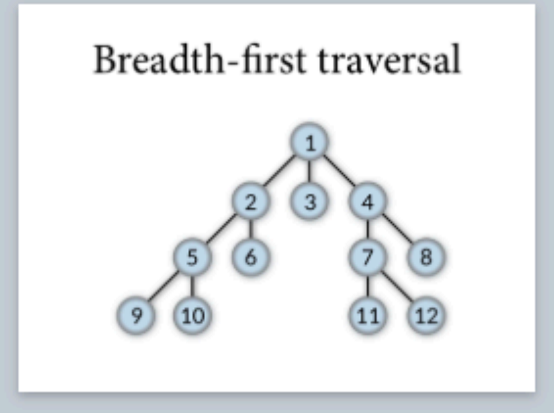
1. Forces you to have a "takeaway"
2. Many readers only care about the takeaway, not the technical details
3. For those who want the technical details, the key ideas are still useful as "scaffolding"

56

A confession

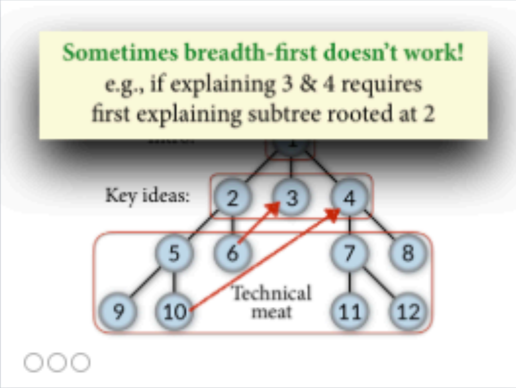
I don't always have a key ideas section.

57



Breadth-first traversal

58

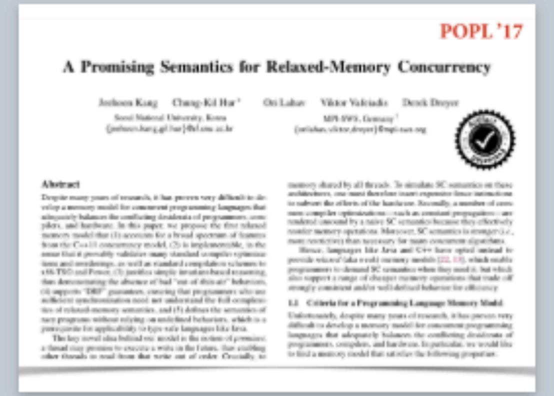


Sometimes breadth-first doesn't work!
e.g., if explaining 3 & 4 requires
first explaining subtree rooted at 2

Key ideas:

Technical meat

59



POPL '17

A Promising Semantics for Relaxed-Memory Concurrency

Jeehoon Kang Chung-Kil Hur Ori Lahav Viktor Vafeiadis Derek Dreyer

Seoul National University, Korea
MPI-SWS, Germany

Abstract

[illegible]

1.1 Criteria for a Programming Language Memory Model

[illegible]

60

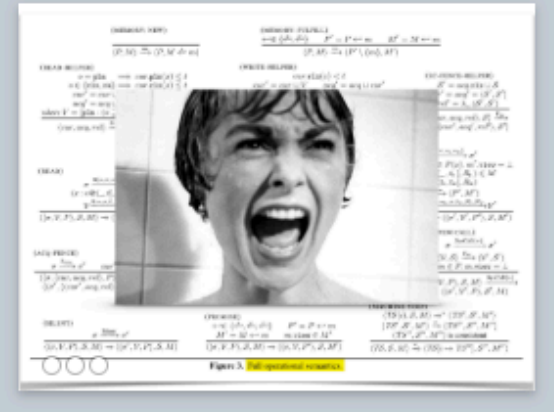
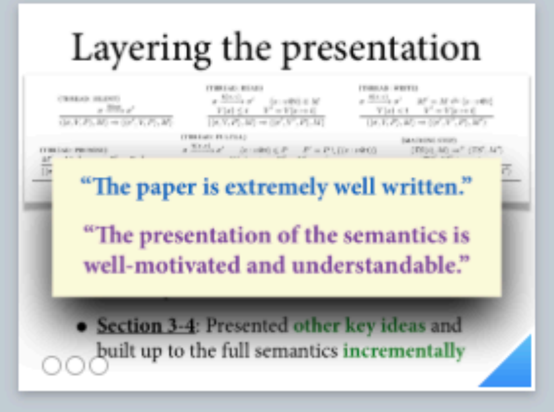


Figure 3. Full operational semantics.

61



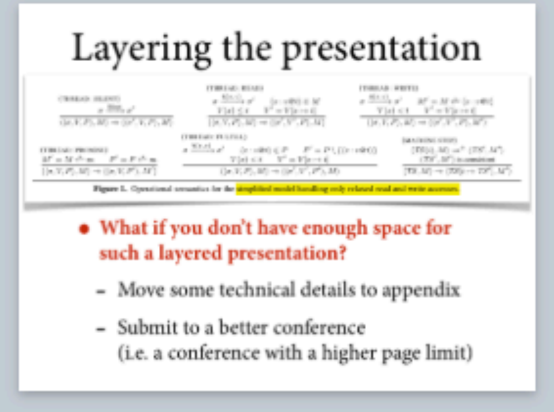
Layering the presentation

"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

62

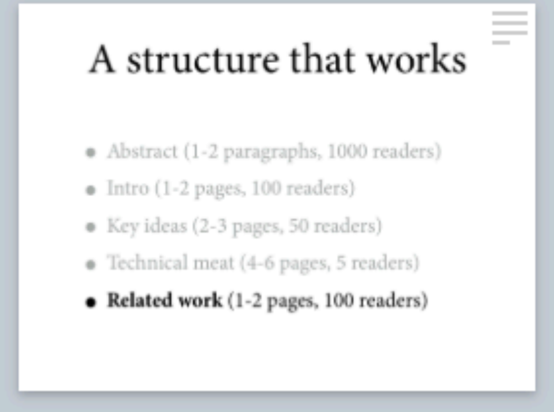


Layering the presentation

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)

63



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)

64

Key takeaways

- **Avoid PowerPoint-itis**
  - Don’t put lots of text on slides just so they are readable independently of the talk

- **Vary the look of the slides**
  - Some text-only slides are fine, but if there are too many in a row, audience falls asleep

Summary of principles

- Talk ≠ Paper
- Intro & key ideas are all you need
- First general problem, then specific problem
- State contributions & follow with key ideas
- Flow via old-to-new
- Coherence via one slide, one point
- Make the focus obvious
- Avoid lots of text & vary the look of slides

Summary of principles

- Talk ≠ Paper
- Intro & key ide
- First genera
- State contribu
- Flow via old-to-new
- Coherence via one slide, one point
- Make the focus obvious
- Avoid lots of text & vary the look of slides

This is what you call avoiding lots of text?

That's all Folks!