How to Bootstrap a Research Project

Ilya Sergey

ilyasergey.net

Yale NUS College

PLMW @ POPL 2019
About myself

MSc  Saint Petersburg State University, 2008
PhD  KU Leuven, 2008-2012

Currently  Associate Professor (tenure-track) at Yale-NUS College & NUS

Previously  Lecturer at University College London
            Postdoc at IMDEA Software Institute
            Software Engineer at JetBrains
Anxiety
Why do a PhD?
Why do a PhD?

Challenge
SECTION A (80 Marks)

Question 1

(i) Find the matrix X for which:

\[
\begin{bmatrix}
  5 & 4 \\
  1 & 1 \\
\end{bmatrix}
\begin{bmatrix}
  1 \\
  -2 \\
\end{bmatrix}
\]

(ii) Solve for x, if:

\[
\tan(\cos^{-1} x) = \frac{2}{\sqrt{3}}
\]

(iii) Prove that the line \(2x - 3y = 9\) touches the conics \(y^2 = -8x\). Also, find the point of contact.

(iv) Using L'Hospital's Rule, evaluate:

\[
\lim_{x \to 0} \left( \frac{1}{x^2} - \frac{\cos x}{x} \right)
\]

(v) Evaluate:

\[
\int \tan^3 x \, dx
\]

(vi) Using properties of definite integrals, evaluate:

\[
\int_0^{\pi/2} \frac{\sin x - \cos x}{1 + \sin x \cos x} \, dx
\]

(vii) The two lines of regressions are \(x + 2y - 5 = 0\) and \(2x + 3y - 8 = 0\) and the variance of \(x\) is 12. Find the variance of \(y\) and the coefficient of correlation.

(viii) Express \(\frac{2 + i}{(1+i)(1-2i)}\) in the form of \(a + ib\). Find its modulus and argument.
More like this kind of Challenge

Millennium Problems

Yang–Mills and Mass Gap
Experiment and computer simulations suggest the existence of a “mass gap” in the solution to the quantum versions of the Yang-Mills equations. But no proof of this property is known.

Riemann Hypothesis
The prime number theorem determines the average distribution of the primes. The Riemann hypothesis tells us about the deviation from the average. Formulated in Riemann’s 1859 paper, it asserts that all the ‘non-obvious’ zeros of the zeta function are complex numbers with real part 1/2.

P vs NP Problem
If it is easy to check that a solution to a problem is correct, is it also easy to solve the problem? This is the essence of the P vs NP question. Typical of the NP problems is that of the Hamiltonian Path Problem: given N cities to visit, how can one do this without visiting a city twice? If you give me a solution, I can easily check that it is correct. But I cannot so easily find a solution.

Navier–Stokes Equation
This is the equation which governs the flow of fluids such as water and air. However, there is no proof for the most basic questions one can ask: do solutions exist, and are they unique? Why ask for a proof? Because a proof gives not only certitude, but also understanding.

Hodge Conjecture
The answer to this conjecture determines how much of the topology of the solution set of a system of algebraic equations can be defined in terms of further algebraic equations. The Hodge conjecture is known in certain special cases, e.g., when the solution set has dimension less than four. But in dimension four it is unknown.

http://www.claymath.org/millennium-problems
What is “?”
When did it go wrong?

• Need to find your own project
• Lost interest in a given project
• Got scooped
• Writer’s Block
Research Limbo

- Is what I’m about to do relevant?
- What if it won’t work?
- Will it scale for more project ideas?
- I’m not as productive as my labmates.
- I don’t see the final goal of what I’m about to do.
You know what to do
Blame Others

- your advisor
- your research environment
- your officemates
- your parents
- the pizza delivery guys
Escaping Research Limbo
Big Picture

Matt Might. *The illustrated guide to a Ph.D.*
http://matt.might.net/articles/phd-school-in-pictures/
### Tools and Techniques
- Type Theory
- Semantics
- Abstract Interpretation
- SMT and Model Checking
- Logics and Proof Assistants
- Program Synthesis, *etc*

### Application Domain
- Machine Learning
- Security and Privacy
- Quantum Computation
- Processor Architecture
- Resource Consumption
- Web, *etc*
One researcher’s **Technique** is another researcher’s **Domain**
One researcher’s Technique is another researcher’s Domain

14:00 - 14:45

*Technical Talk: How to Think about Types*
Frank Pfenning Carnegie Mellon University, USA

14:45 - 15:30

*Technical Talk: What Is Programming Languages Research?*
Michael Hicks University of Maryland, College Park
This year at POPL

- Pretend Synchrony: Synchronous Verification of Asynchronous Distributed Programs
- A Separation Logic for Concurrent Randomized Programs
- Bayesian Synthesis of Probabilistic Programs for Automatic Data Modeling
- An Abstract Domain for Certifying Neural Networks
- Game Semantics for Quantum Programming
- Type-Driven Gradual Security with References
- ISA Semantics for ARMv8-A, RISC-V, and CHERI-MIPS
- JaVerT 2.0: Compositional Symbolic Execution for JavaScript
- LWeb: Information Flow Security for Multi-Tier Web Applications
Escaping Research Limbo

Step 1: Read
Escaping Research Limbo

- Is what I’m about to do relevant?
- What if it won’t work?
- Will it scale for more project ideas?
- I’m not as productive as my labmates.
- I don’t see the final goal of what I’m about to do.
Escaping Research Limbo

- Is what I’m about to do relevant?
  - What if it won’t work?
  - Will it scale for more project ideas?
  - I’m not as productive as my labmates.
  - I don’t see the final goal of what I’m about to do.
Escaping Research Limbo

- Is what I’m about to do relevant?
  - What if it won’t work?
    - Will it scale for more project ideas?
    - I’m not as productive as my labmates.
    - I don’t see the final goal of what I’m about to do.
Finding the Path
There is no royal road to geometry.
Euclid

Research is from “search”.
Aleks Nanevski

Move Fast and Break Things.
© facebook
Escaping Research Limbo

Step 2: Hack
Pick your Favourite Tools
A typical POPL paper

• Introduction
• Overview
• Theory
• Mechanisation / Implementation
• Case Studies / Evaluation
• Related Work and Conclusion
A typical POPL paper

- Introduction
- Overview
  - Theory
  - Mechanisation / Implementation
  - Case Studies / Evaluation
- Related Work and Conclusion
The Scientific Method

Theory

Implementation

Evaluation
Implementation
This paper describes a deductive approach to synthesizing imperative programs with pointers from declarative specifications expressed in Separation Logic. Our synthesis algorithm takes as input a pair of assertions—
“Hello World” works!

Implementation

Distilling the theory

Fancy Case Studies

Optimisations

POPL deadline
Escaping Research Limbo

- Is what I’m about to do relevant?

  - What if it won’t work?
  
    - Will it scale for more project ideas?
    - I’m not as productive as my labmates.
    - I don’t see the final goal of what I’m about to do.
Escaping Research Limbo

- Is what I’m about to do relevant?
- What if it won’t work?
- Will it scale for more project ideas?
- I’m not as productive as my labmates.
- I don’t see the final goal of what I’m about to do.
Escaping Research Limbo

- Is what I’m about to do relevant?
- What if it won’t work?
- Will it scale for more project ideas?
- I’m not as productive as my labmates.
- I don’t see the final goal of what I’m about to do.
Step 3: Look for Shortcomings
opportunity
The Virtue of PL Research

Implementation to Solve a Domain Problem

too messy
too brittle
too inefficient

Can we make it better?

A novel technique for better implementations.
In praise of Incrementality

Bertrand Meyer. *One cheer for incremental research*
https://bertrandmeyer.com/2009/08/10/one-cheer-for-incremental-research/
Story One

“Can we make Static Analyses less complex?”
Abstracting Abstract Machines

David Van Horn *
Northeastern University
dvanhorn@ccs.neu.edu

Matthew Might
University of Utah
might@cs.utah.edu

ICFP’10

Introspective Pushdown Analysis of Higher-Order Programs

Christopher Earl
University of Utah
cwearl@cs.utah.edu

Ilya Sergey
KU Leuven
ilya.sergey@cs.kuleuven.be

Matthew Might
University of Utah
might@cs.utah.edu

David Van Horn
Northeastern University
dvanhorn@ccs.neu.edu

ICFP’12

Monadic Abstract Interpreters

Ilya Sergey
IMDEA Software Institute, Spain
ilya.sergey@imdea.org

Dominique Devriese
iMinds – DistriNet, KU Leuven, Belgium
dominique.devriese@cs.kuleuven.be

Matthew Might
University of Utah, USA
might@cs.utah.edu

Jan Midtgaard
Aarhus University, Denmark
jmi@cs.au.dk

David Darais
Harvard University, USA
darais@seas.harvard.edu

PLDI’13

Dave Clarke
Frank Piessens
iMinds – DistriNet, KU Leuven, Belgium
(firstname.lastname)@cs.kuleuven.be

See David Darais’ works for the follow-ups.

too inefficient

lots of code duplication

Can we make Static Analyses less complex?
Story Two

“Can we unify existing Concurrency Logics?”
Concurrent separation logic is essentially as if it were atomic, even if it is not. Although the semantics of sequentially consistent (SC) concurrency is simple to define, that does not mean it is easy to reason about. Several newer extensions to CSL \[1\], built in as primitive in prior logics. By this we mean that there is an inherent stratification: the work of Owicki and Gries \[2\] has tied assertions to a "conditional critical region" construct for synchronizing multiple threads. This approach is no longer needed in the presence of fine-grained concurrency, for which logics must support the ability to store arbitrary higher-order data, taking recursivity and references into account.

How do we use it?

In the last decade, there has been tremendous progress on program verification, especially for fine-grained concurrent programs. This is the first formalization of a binary logical relation that supports all of these different kinds of resources (they are instrumenting a program with updates to "ghost" (or "auxiliary") state). Several newer extensions to CSL \[1\], \[3\] and \[4\] were tied to a "conditional critical region" construct for synchronizing multiple threads. This approach is no longer needed in the presence of fine-grained concurrency, for which logics must support the ability to store arbitrary higher-order data, taking recursivity and references into account.

POPL’17
Interactive Proofs in Higher-Order Concurrent Separation Logic

- Robbert Krebbers *
  Delft University of Technology, The Netherlands
  mail@robbertkrebbers.nl

- Amin Timany
  imec-DistriNet, KU Leuven, Belgium
  amin.timany@cs.kuleuven.be

- Lars Birkedal
  Aarhus University, Denmark
  birkedal@cs.au.dk

POPL’16
Iris: Monoids and Invariants as an Orthogonal Basis for Concurrency Reasoning

- Ralf Jung
  MPI-SWS, Germany
  jung@mpi-sws.org

- Robbert Krebbers
  Aarhus University, Denmark
  mail@robbertkrebbers.nl

- Lars Birkedal
  Aarhus University, Denmark
  birkedal@cs.au.dk

- Derek Dreyer
  MPI-SWS, Germany
  dreyer@mpi-sws.org

ICFP’16
Higher-Order Ghost State

- Ralf Jung
  MPI-SWS, Germany
  jung@mpi-sws.org

- Robbert Krebbers
  Aarhus University, Denmark
  mail@robbertkrebbers.nl

- Lars Birkedal
  Aarhus University, Denmark
  birkedal@cs.au.dk

- Derek Dreyer
  MPI-SWS, Germany
  dreyer@mpi-sws.org

Can we unify existing Concurrency Logics?

Proofs are difficult to manage

In the last decade, there has been tremendous progress on program verification, especially for fine-grained concurrent programs. This is the first formalization of a binary logical relation that supports all of these different kinds of resources (they are instrumenting a program with updates to "ghost" (or "auxiliary") state). Several newer extensions to CSL \[1\], \[3\] and \[4\] were tied to a "conditional critical region" construct for synchronizing multiple threads. This approach is no longer needed in the presence of fine-grained concurrency, for which logics must support the ability to store arbitrary higher-order data, taking recursivity and references into account.

How do we use it?

In the last decade, there has been tremendous progress on program verification, especially for fine-grained concurrent programs. This is the first formalization of a binary logical relation that supports all of these different kinds of resources (they are instrumenting a program with updates to "ghost" (or "auxiliary") state). Several newer extensions to CSL \[1\], \[3\] and \[4\] were tied to a "conditional critical region" construct for synchronizing multiple threads. This approach is no longer needed in the presence of fine-grained concurrency, for which logics must support the ability to store arbitrary higher-order data, taking recursivity and references into account.

Can we unify existing Concurrency Logics?

Proofs are difficult to manage

In the last decade, there has been tremendous progress on program verification, especially for fine-grained concurrent programs. This is the first formalization of a binary logical relation that supports all of these different kinds of resources (they are instrumenting a program with updates to "ghost" (or "auxiliary") state). Several newer extensions to CSL \[1\], \[3\] and \[4\] were tied to a "conditional critical region" construct for synchronizing multiple threads. This approach is no longer needed in the presence of fine-grained concurrency, for which logics must support the ability to store arbitrary higher-order data, taking recursivity and references into account.

How do we use it?

In the last decade, there has been tremendous progress on program verification, especially for fine-grained concurrent programs. This is the first formalization of a binary logical relation that supports all of these different kinds of resources (they are instrumenting a program with updates to "ghost" (or "auxiliary") state). Several newer extensions to CSL \[1\], \[3\] and \[4\] were tied to a "conditional critical region" construct for synchronizing multiple threads. This approach is no longer needed in the presence of fine-grained concurrency, for which logics must support the ability to store arbitrary higher-order data, taking recursivity and references into account.

Can we unify existing Concurrency Logics?

Proofs are difficult to manage
Standing on the Shoulders of Giants

16:30 - 17:30  Panel: How to Do Great PL Research

Vasco Vasconcelos LASIGE, Faculty of Sciences, University of Lisbon, Deepak Garg Max Planck Institute for Software Systems, Philippa Gardner Imperial College London, Atsushi Igarashi Kyoto University, Japan, Neelakantan R. Krishnaswami Computer Laboratory, University of Cambridge
Escaping Research Limbo

- Is what I’m about to do relevant?
- What if it won’t work?
- Will it scale for more project ideas?
  - I’m not as productive as my labmates.
  - I don’t see the final goal of what I’m about to do.
Escaping Research Limbo

• Is what I’m about to do relevant?
• What if it won’t work?
• Will it scale for more project ideas?

• I’m not as productive as my labmates.
• I don’t see the final goal of what I’m about to do.
Escaping Research Limbo

• Is what I’m about to do relevant?
• What if it won’t work?
• Will it scale for more project ideas?
• I’m not as productive as my labmates.
• I don’t see the final goal of what I’m about to do.
POPL  CAV  PLDI  POPL again

your officemate now

you now
two years ago

uphill battle

productivity plateau

when you started

now

learning new techniques/domain
Escaping Research Limbo

Step 4: Document your results

milestone papers → flagship conference

workshop → symposium
My first Highs and Lows

Inspired by Greg Morrisett’s *Highs and Lows of a Language Researcher*. PLMW@POPL’16

---

**Project 1 (PhD)**
- IPL’11
- ESOP’12
- ESOP’15

**Project 2 (postdoc)**
- PLDI’15
- ESOP’14
- ESOP’15
- OOPSLA’16
- ECOOP’17

---

Learning about concurrency, Separation Logic, and Coq
Escaping Research Limbo

• Is what I’m about to do relevant?
• What if it won’t work?
• Will it scale for more project ideas?
• I’m not as productive as my labmates.
• I don’t see the final goal of what I’m about to do.
Escaping Research Limbo

• Is what I’m about to do relevant?
• What if it won’t work?
• Will it scale for more project ideas?
• I’m not as productive as my labmates.

• I don’t see the final goal of what I’m about to do.
Escaping Research Limbo

• Is what I’m about to do relevant?

• What if it won’t work?

• Will it scale for more project ideas?

• I’m not as productive as my labmates.

• I don’t see the final goal of what I’m about to do.
You have acquired skills

They are reusable
To Take Away

• Starting a new project is always scary!

• To escape the *Research Limbo*:
  Read, Hack, Find shortcomings, Write

Many thanks to my fellow travellers: Dave Clarke, Aleks Nanevski, Olivier Danvy, Matt Might, David Van Horn, Simon Peyton Jones, Dimitrios Vytiniotis, Dominique Devriese, Aquinas Hobor, Nadia Polikarpova, Jan Midtgaard, Peter O’Hearn, Nikos Gorogiannis, Álvaro García Pérez, Anindya Banerjee, Zach Tatlock, Germán Delbianco, David Darais, Anton Podkopaev, Kristoffer Just Andersen, Maria A Schett, George Pîrlea, Kiran Gopinathan, and James R. Wilcox.

Thank you and good luck!