Background in category theory (and functional programming) in form of a timeline
- 2005/2006: watched SICP lecture videos at MIT from 1986
- 2006: first learned of category theory from Mark Chu-Carrol's blog
- 2007: learned Haskell for the first time
- 2008/2009: watched Catsters Youtube videos by Eugenia Cheng et al
- 2009: read a few papers on categorical view on functional programming by Uustalu and Vene
- 2009: taught a very informal study group on categories and functional programming in Helsinki
- 2010: started graduate school at Penn, studied category theory in algebra course
- 2011: started learning algebraic geometry for my PhD
- 2011: Vladimir Voevodsky gave a talk at Penn, I got curious of type theory as foundations of mathematics
- 2015: took part in study group on homotopy type theory
- 2015: graduated with a PhD, started a (non-academic) job
- 2018: got into reading about applied category theory
- 2018/2019: watched Spivak & Fong lectures at MIT, the ones that are available online
- 2019: reading parts of the Seven sketches book, chapters 1 and 7
- 2019: heard of ACT school, decided to apply

PhD thesis
It was submitted June 18th 2015, I officially graduated in August 2015
Title: Lifting Problems and Their Independence of the Coefficient Field
(pdf available at http://www.math.upenn.edu/grad/dissertations/AstrandThesis.pdf)
Advisor: Florian Pop
Short summary:
    The thesis uses model theory of fields to prove a slightly stronger
    version of Oort conjecture on lifting curves from characteristic p to characteristic 0.

Project preference
(These are mild preferences, all of the projects sound quite interesting)
1. Mehrnoosh Sadrzadeh: Discourse using categorical models
2. Tobias Fritz: Partial evaluation, bar construction
3. Miriam Backens: Quantum ZX-calculus
4. Bartosz Milewski: Optics, profunctors
5. David Spivak: Autopoiesis
6. Pieter Hofstra: Turing categories, complexity

Traveling to Oxford
Oxford is a short train ride away from me, so I can commit to coming there. For the train
tickets I don't need any funding, and I can stay there for at least a week, probably two.

Letter of recommendation
You should have received an email from Florian Pop, who was my thesis advisor at Penn.

Matti Åstrand

Phone number: +44-7521-047687, Email: mattiastr@gmail.com

Experience

- Software Engineer at **Smarkets** betting exchange starting from December 2017
  - Developing microservices based system mainly in Python
  - Implementing systematic market making strategies for sports betting
- Associate at **Goldman Sachs** from July 2015 to September 2017, working as a **Desk strat in Structured Credit Trading**.
  - Developing pricing and risk calculations for credit derivatives using GS proprietary Slang language
  - Producing many (automated) reports, both for real time risk monitoring and overnight P&L and regulatory reports
  - Maintaining a large number of batch processes in a distributed job scheduling system
- Teaching at **University of Pennsylvania Math department**: 5 semesters as a Teaching Assistant and one semester as a Lecturer. I was given two **Teaching Awards** for Good Teaching
- **Math tutoring** (both volunteer and paid) starting from 2010 on every level from third grade to advanced undergraduate courses
- Research Assistant at **Helsinki Institute for Information Technology** from January 2009 to December 2009
  - Co-authored three papers in Computer Science theory, two of which were published in conferences
  - Wrote a C program to simulate and test the performance of local distributed algorithms
- Trainee at **Nokia Research Center** from September 2003 to August 2005
  - Implemented mobile applications in Java for testing and demoing some new API's developed by Nokia and others
  - Wrote applications for a 3D graphics library JSR 184, and a Bluetooth communications library
- GitHub account `github.com/mattiast`
- I have taken part in various problem solving/programming competitions online:
  189 problems in `projecteuler.net`, 104 problems in `codeeval.com` using mostly Haskell

Education

- **PhD in Mathematics**, University of Pennsylvania, August 2015
- **Master of Science in Mathematics**, University of Helsinki, June 2009
- **Bachelor of Science in Mathematics** (minor in CS and Physics), University of Helsinki, February 2009
- I have watched lots of online courses from universities such as MIT, Berkeley, Stanford, Yale, Harvard on topics such as Mathematics, CS, Machine learning, Finance, Electrical engineering and Psychology

Skills

- Programming languages: Python, Haskell, C, Elm, Rust
- Software: LaTeX, Git, Vim, Docker, Kubernetes
- I have used Linux since 2003
- Language skills: Finnish (native), English (fluent), Estonian (fluent), Swedish (basic), Russian (basic)

Publications

- **Matti Åstrand** and Jukka Suomela, "Fast distributed approximation algorithms for vertex cover and set cover in anonymous networks", *22nd ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)*, Santorini, Greece, June 2010.
- **Matti Åstrand**, Patrik Floréen, Valentin Polishchuk, Joel Rybicki, Jukka Suomela, and Jara Uitto, "A local 2-approximation algorithm for the vertex cover problem", *23rd International Symposium on Distributed Computing (DISC)*, Elche, Spain, September 2009.
- **Matti Åstrand**, Valentin Polishchuk, Joel Rybicki, Jukka Suomela and Jara Uitto, "Local algorithms in (weakly) coloured graphs", 2010.

Awards and achievements

- Benjamin Franklin Fellowship from University of Pennsylvania: full funding for 4 years of graduate studies
- Stipend of $10000 from **American Scandinavian Foundation** in 2010 for graduate studies
- First Prize in the **International Mathematics Competition for University Students** in 2008
- Second Prize in the **International Mathematics Competition for University Students** in 2009
- Honorable Mention in the **International Physics Olympiad** in 2005
- Stipend of 1000€ from **The Federation of Finnish Technology Industries** for getting full score in Mathematics Matriculation Examination in 2005

U N I V E R S I T Y *of* P E N N S Y L V A N I A

Department of Mathematics DRL, 209 S 33rd Street, Philadelphia, PA 19104

Florian Pop
Samuel D. Schack Professor of Algebra
Professor of Mathematics

Tel. +1-215-898-5971
pop@math.upenn.edu
www.math.upenn.edu/~pop

Jan 2019

To whom it may concern:

I am writing to enthusiastically recommend Dr. Matti Astrand in **strongest possible terms** for the Adjoint School ACT 2019.

Matti Astrand completed his Ph.D. in the Graduate Program in Mathematics at the University of Pennsylvania in the spring of 2015, with a thesis in the area of arithmetic algebraic geometry and model theory. In his thesis Matti solved a very difficult open problem concerning the so called *Oort groups,* which was part of a wider research group effort funded by the NSF (National Science Foundation), involving scientists at three Universities in the U.S., and collaborators from the U.S. and Europe. Matti's thesis result was much higher than the usual level of a Ph.D. thesis, and I can say without hesitation that he was at the top of his generation of graduate students.

Matti Astrand was as well an enthusiastic teaching assistant and a great communicator of mathematical content, and that combined with his great thesis result constituted a strong basis for a career in academia. On the other hand, Matti had a lively curiosity, being interested both in research in other areas of mathematics and computer science, and in applications of pure math in science and everyday life.

I am glad to conclude by giving Matti Astrand my enthusiastic **strongest possible recommendation** for the Adjoint School ACT 2019. I am convinced that he will be an active participant and contribute to the success of the sctivity, and visiting the school will be very profitable for him.

Sincerely,

Florian Pop
(*Samuel D. Schack Professor of Algebra* / Professor of Mathematics)

Prelude
The topic of Applied category theory ties together (at least) three themes that have shaped
my interests and career. These are, in roughly chronological order:
1. pure mathematics
2. functional programming
3. pragmatic software design

Pure mathematics
Category theory was first of interest to me in context of algebra, as a way to abstract
specifics of various kinds of homomorphisms.

I had some fun thinking about forgetful functors (e.g. from monoids to sets) and trying to construct
left adjoints for them. This is a way to generate concepts of "free" groups, monoids etc.
It is easy to see the value of using categorical concepts as a language in pure mathematics.

Functional programming
My first exposure to functional programming was watching MIT lectures on their course 6.001
called "Structure and interpretation of computer programs". The lectures were from 1986.

The SICP course used the programming language Scheme. Later I discovered Haskell,
and to this day I find lots of joy from writing it.
Discovering functional programming sparked my interest towards coding again.
Before that I had only used imperative languages, which I didn't find as elegant.

Reading papers by Uustalu and Vene was another source of ideas. Being able
to combine programming and abstract mathematics was impressive, and gave me some reason to
think that category theory may be useful more widely than in pure mathematics.

Software design principles
Already in the SICP course one theme that came up over and over was thinking about system design with three aspects:
- primitive elements
- means of combination: how can those elements be combined into more complicated elements?
- means of abstraction: can we abstract the combined elements and use them as if they were primitive?

In software design there are many nice properties for systems to have.
One desirable adjective is **composable**: things can be put together to build more complicated things.
Another one is **orthogonal**: we can make various combinations of different parts independently of
each other. These two are quite close to each other in meaning.

Functional programmers often talk about **combinator libraries**, where we start by defining building
blocks, and then ways to combine them.
One good example (in my opinion) is the deep learning library Keras, where you can put layers together
as long as their input and output types and dimensions match. Design of Keras is

 already quite close to a
monoidal category, where the layers are arrows between input and output types.

All of these ideas are related to the notion of compositionality -- they all are
 essentially
saying that parts should be easy to put together in flexible ways.

Conclusion
I see promise in using category theory as one guiding tool for designing and des
cribing systems.
For this reason I'm interested in learning about and contributing to the study o
f applied category
theory.

With my background in both pure mathematics and software development, I think I
can contribute to the
conversation. This would also help my career in giving me insight and experience
 in a novel kind of applied
mathematics.