

**Speaker:** Antonio Filieri

**Title:** Probabilistic programming: a software engineering perspective

**Abstract:** This short seminar will overview some of the essentials of probabilistic programming from a software engineering and programming languages perspective. What makes a language probabilistic? How do we encode Bayesian thinking in a programming system? Can we test a probabilistic program?

**Short bio:**

Antonio Filieri is a Lecturer in Computing at Imperial College London, UK. His main research interests are in mathematical methods for Software Engineering, particularly applications of probability, statistics, logic, and control theory. His recent work includes exact and approximate methods for probabilistic symbolic execution, incremental verification, quantitative software modeling and verification at runtime, and control-theoretical software adaptation. <https://antonio.filieri.name> .

**# Reading list (optional)**

## Basic definitions:

Probabilistic Programming

Gordon, Henzinger, Nori, Rajamani

<https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/fose-icse2014.pdf>

Basic practical course, in the browser:

<https://probmods.org/>

## Some design and implementation strategies for probabilistic programming systems:

Lightweight implementations of probabilistic programming languages via transformational compilation

Wingate, Stuhlmüller, and Goodman

<http://proceedings.mlr.press/v15/wingate11a/wingate11a.pdf>

Uncertain<T>: a first-order type for uncertain data

Bornholt, Mytkowicz, and McKinley

<https://doi.org/10.1145/2541940.2541958>

A Compilation Target for Probabilistic Programming Languages

Paige, Wood

<http://proceedings.mlr.press/v32/paige14.pdf>

Bayesian inference using data flow analysis

Claret, Rajamani, Nori, Gordon, Borgstroem

<https://doi.org/10.1145/2491411.2491423>

Deep probabilistic programming

Tran, Hoffman, Saurous, Brevdo, Murphy, and Blei

<https://arxiv.org/pdf/1701.03757.pdf>

Testing probabilistic programming systems

Dutta, Legunsen, Huang, Misailovic

<https://doi.org/10.1145/3236024.3236057>

## A couple of less "conventional" examples:

Bayesian synthesis of probabilistic programs for automatic data modeling

Saad, Cusumano-Towner, Schaechtle, Rinard, and Mansinghka

<https://doi.org/10.1145/3290350>

Picture: A Probabilistic Programming Language for Scene Perception

Kulkarni, Kohli, Tenenbaum, Mansinghka

<https://doi.org/10.1109/CVPR.2015.7299068>

Etalumis: bringing probabilistic programming to scientific simulators at scale

Baydin et al.

<https://doi.org/10.1145/3295500.3356180>

## On Bayesian thinking, or the philosophy behind probabilistic programming:

How to Grow a Mind: Statistics, Structure, and Abstraction

Tenenbaum, Kemp, Griffiths and Goodman

<https://www.science.org/doi/full/10.1126/science.1192788>

Dicing with the unknown

Hagan

[http://www.stat.columbia.edu/~gelman/stuff\\_for\\_blog/ohagan.pdf](http://www.stat.columbia.edu/~gelman/stuff_for_blog/ohagan.pdf)