# Introduction to Program Synthesis (SS 2025)

## Chapter 0 - Organization and Overview

Dr. rer. nat. Roman Kalkreuth

Chair for AI Methodology (**AIM**), Department of Computer Science,
RWTH Aachen University, Germany


# Chair for AI Methodology (AIM)

- Chairholder: **Professor Holger Hoos**
- Trustworthy, human-centred AI
  - ⇝ AI for good, AI for all
- Methodological, technological advances of AI methods, systems and tools for benefit of society and humanity
  - ⇝ Automated machine learning (AutoML)
  - ⇝ Neural network verification
  - ⇝ Program synthesis
  - ⇝ AI for arts
- Part of the international ADA research group
  - ⇝ LIACS, Leiden University, NL
  - ⇝ University of British Columbia, Canada

# About me

**Current Role:** Assistant Professor (Akademischer Rat)

**Research Objectives:** Program synthesis with graph-based genetic programming, Benchmarking of randomized search heuristics

# Scope

- One lecture per week
- One exercise per week (?)
  - ⇝ Exercise sheets with mandatory submissions (via mail)
  - ⇝ Hybrid exercises that cover theoretical and practical scope
  - ⇝ Literature discussions (in-person)
- Written exam at the end of the course
  - ⇝ More information will be announced

# Scope

- Lecture $\rightarrow$ Teaching the theoretical foundations
  - $\rightsquigarrow$ **Wednesday 12:30 - 14:00**
- Exercise $\rightarrow$ Independent recapitulation of the theory
  - $\rightsquigarrow$ Recapitulation and deepening of the theory
  - $\rightsquigarrow$ Implementation, evaluation and analysis of PS methods and related foundations

## Lecture concepts

- Learning by demonstration $\rightarrow$ **direct learning** with **examples** and **live demonstrations**
- Joint discussion and review of **papers** that are related to program synthesis
- Consideration of recent developments

# Exercise

- Location $\rightarrow$ **RWTH Chair for AI Methodlogy, Theaterstraße 35-39 (third floor)**
  - $\rightsquigarrow$ https://maps.app.goo.gl/m8MHbzzeC1QgzS378
- Implementations should be written in **Python**
- Processing and submission of the exercise sheets will be done in **group work**
  - $\rightsquigarrow$ Group size should be between 2 and 4 people

# Exercise: Code of Conduct

- Treat all course members with respect
- Behave professionally. harassment and sexist, racist or marginalising comments will not be tolerated
- Unacceptable behaviour generally includes: harassing, abusive, discriminatory, intimidating, degrading and humiliating behaviour
- This also holds for online communication

# Curriculum (overview)

- **Introduction**
  - ⇝ Scope and History of PS
- **Foundations**
  - ⇝ Computer Programs
  - ⇝ Programming Paradigms and Languages
  - ⇝ Machine Learning
  - ⇝ Optimization
- **General Principles of PS**
  - ⇝ Problem Statement and Analysis
  - ⇝ Search Spaces
- **Traditional Search Methodologies**
  - ⇝ Enumerative Search
  - ⇝ Inductive/Deductive Search
  - ⇝ Stochastic Search
- **Deep Learning-based Methodologies**
  - ⇝ Neural Program Synthesis
  - ⇝ GenAI: Transformer and LLMs

# Course objectives

- Developing a **solid understanding** of the **fundamentals** required for the **application of PS**
- Obtaining an overview of **traditional** and **modern methods** of PS
- Gaining an understanding of which methods are **relevant** for **program synthesis** today
  - $\rightsquigarrow$ Getting to know the **shortcomings** and drawbacks of **contemporary methods**

# Requirements

- Fundamental programming skills in **Python**
- Fundamental understanding of **data structures**, **numerical analysis** and **linear algebra**

# Resources

- GitHub repository
  - ⇝ Lecture slides
  - ⇝ Examples written in C or Python
  - ⇝ Will be publicly available

https://github.com/RomanKalkreuth/program-synthesis-lecture

## Consultation

- By appointment via mail: kalkreuth@aim.rwth-aachen.de
- Office address: **Theaterstraße 35-39 (third floor)**

# Feedback

- Relatively new course
- Feedback is crucial for the improvement of the course and is always appreciated

# Literature

- Literature is still rather thin
- A comprehensive book on program synthesis ("*Bible*") is still missing
- Studying key publications are therefore more the way to go
- Additionally various surveys are available

# Literature I


[GPS17] Sumit Gulwani, Oleksandr Polozov, and Rishabh Singh. "Program Synthesis". In: Found. Trends Program. Lang. 4.1-2 (2017), pp. 1–119. DOI: 10.1561/2500000010. URL: https://doi.org/10.1561/2500000010.

[SGF10] Saurabh Srivastava, Sumit Gulwani, and Jeffrey S. Foster. "From Program Verification to Program Synthesis". In: Proceedings of the 37th Annual ACM SIGPLAN-SIGACT Symposium on Pri POPL '10. Madrid, Spain: ACM, 2010, pp. 313–326. ISBN: 978-1-60558-479-9. DOI: 10.1145/1706299.1706337. URL: http://doi.acm.org/10.1145/1706299.1706337.

[HS00] Gary D. Hachtel and Fabio Somenzi. Logic Synthesis and Verification Algorithms. 1st. USA: Kluwer Academic Publishers, 2000. ISBN: 0792397460.

# Literature II


[Kit10] Emanuel Kitzelmann. "Inductive Programming: A Survey of Program Synthesis Techniques". In: Approaches and Applications of Inductive Programming. Ed. by Ute Schmid, Emanuel Kitzelmann, and Rinus Plasmeijer. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 50–73. ISBN: 978-3-642-11931-6.

[Kre98] Christoph Kreitz. "Program Synthesis". In: Automated Deduction — A Basis for Applications: Volume III Applications. Ed. by Wolfgang Bibel and Peter H. Schmitt. Dordrecht: Springer Netherlands, 1998, pp. 105–134. ISBN: 978-94-017-0437-3. DOI: 10.1007/978-94-017-0437-3_5. URL: https://doi.org/10.1007/978-94-017-0437-3_5.

[Koz92] John R. Koza. Genetic Programming: On the Programming of Computers by Means of Natu Cambridge, MA, USA: MIT Press, 1992. ISBN: 0-262-11170-5. URL: http://mitpress.mit.edu/books/genetic-programming.