

# Seminar/Proseminar

# Modellbasierte Entwicklung

Sommersemester 2021

# Organisation

**Bachelor: Proseminar      Master: Seminar**

**Wissenschaftliche Präsentation** zum gewählten Thema im Rahmen einer zentralen Veranstaltung am Ende des Semesters.

- Proseminar: 20 Minuten Vortrag + 10 Minuten Diskussion
- Seminar: 30 Minuten Vortrag + 15 Minuten Diskussion

**Schriftliche Ausarbeitung** des gewählten Themas, abzugeben ca. 4 Wochen nach dem Vortragstermin.

- Proseminar: 6 Seiten (ACM Style)
- Seminar: 10 Seiten (ACM Style)

# Meilensteine

1. Besprechung der Literatur und erster Entwurf der Gliederung der Ausarbeitung (ca. 4 Wochen nach der Einführungsveranstaltung)
2. Abgabe eines ersten Zwischenstandes der Ausarbeitung und Anfertigen von Reviews für die Ausarbeitungen anderer Teilnehmer (ca. 4 Wochen vor dem Vortragstermin)
  - Proseminar: 1 Review
  - Seminar: 2 Reviews
3. Abgabe eines ersten Präsentationsentwurfs und Probevortrag (ca. 2 Wochen vor dem Vortragsbeginn)

## LLVM: A Compilation Framework for Lifelong Program Analysis & Transformation

Chris Lattner      Vikram Adve  
University of Illinois at Urbana-Champaign  
{lattner,vadve}@cs.uiuc.edu  
<http://llvm.cs.uiuc.edu/>

## Emscripten: An LLVM-to-JavaScript Compiler

Alon Zakai  
Mozilla  
azakai@mozilla.com

Software landscape and application visualization for system comprehension with ExplorViz

Florian Fittkau<sup>1,\*</sup>, Alexander Krause, Wilhelm Hasselbring

*Software Engineering Group, Kiel University, D-24098 Kiel, Germany*

## Comparing Trace Visualizations for Program Comprehension through Controlled Experiments

Florian Fittkau, Santje Finke, Wilhelm Hasselbring, and Jan Waller

Software Engineering Group, Kiel University, Kiel, Germany

Email: {ffi, sfi, wha, jwa}@informatik.uni-kiel.de

## **Trace Visualization for Program Comprehension: A Controlled Experiment**

Bas Cornelissen, Andy Zaidman, Arie van Deursen  
Delft University of Technology  
The Netherlands

{s.g.m.cornelissen, a.e.zaidman, arie.vandeursen}@tudelft.nl

Bart van Rompaey  
University of Antwerp  
Belgium

bart.vanrompaey2@ua.ac.be

## Comparing Trace Visualizations for Program Comprehension through Controlled Experiments

Florian Fittkau, Santje Finke, Wilhelm Hasselbring, and Jan Waller  
Software Engineering Group, Kiel University, Kiel, Germany  
Email: {ffi, sfi, wha, jwa}@informatik.uni-kiel.de

## Template-based program verification and program synthesis

Saurabh Srivastava · Sumit Gulwani ·  
Jeffrey S. Foster

### Dimensions in Program Synthesis

[Invited Talk Paper]

Sumit Gulwani

Microsoft Research, Redmond, WA, USA  
sumitg@microsoft.com

## FlashMeta: A Framework for Inductive Program Synthesis

Oleksandr Polozov

University of Washington, USA  
polozov@cs.washington.edu

Sumit Gulwani

Microsoft Research, USA  
sumitg@microsoft.com

## Dimensions in Program Synthesis

[Invited Talk Paper]

Sumit Gulwani

Microsoft Research, Redmond, WA, USA  
sumitg@microsoft.com



## LICCA: A Tool for Cross-Language Clone Detection

Tijana Vislavski\*, Gordana Rakić\*, Nicolás Cardozo† and Zoran Budimac\*

\* University of Novi Sad, Faculty of Sciences - Novi Sad, Serbia

{tijana.vislavski, gordana.rakic, zjb}@dmi.uns.ac.rs

† Universidad de los Andes, Systems and Computing Engineering Department - Bogotá, Colombia

n.cardozo@uniandes.edu.co

## **NICAD: Accurate Detection of Near-Miss Intentional Clones Using Flexible Pretty-Printing and Code Normalization**

Chanchal K. Roy and James R. Cordy  
School of Computing, Queen's University  
Kingston, ON, Canada K7L 3N6  
{croy, cordy}@cs.queensu.ca

## A Systematic Review on Code Clone Detection

**QURAT UL AIN, WASI HAIDER BUTT<sup>ID</sup>, MUHAMMAD WASEEM ANWAR<sup>ID</sup>,  
FAROOQUE AZAM, AND BILAL MAQBOOL**

Department of Computer and Software Engineering, College of Electrical and Mechanical Engineering, National University of Sciences and Technology (NUST), Islamabad 44000, Pakistan

Corresponding author: Muhammad Waseem Anwar (waseemanwar@ceme.nust.edu.pk)

## A Choice of Variational Stacks: Exploring Variational Data Structures

Meng Meng,<sup>1</sup> Jens Meinicke,<sup>2,3</sup> Chu-Pan Wong,<sup>2</sup> Eric Walkingshaw,<sup>1</sup> Christian Kästner<sup>2</sup>

<sup>1</sup>Oregon State University, Corvallis, OR, USA

<sup>2</sup>Carnegie Mellon University, Pittsburgh, PA, USA

<sup>3</sup>University of Magdeburg, Magdeburg, Germany

## Variational Data Structures: Exploring Tradeoffs in Computing with Variability

Eric Walkingshaw  
University of Marburg  
Germany

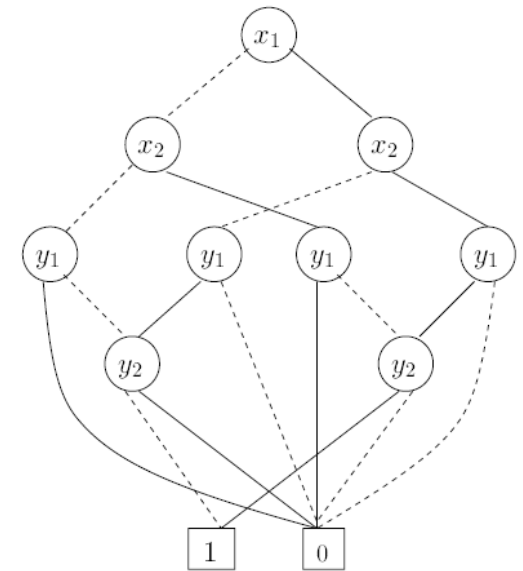
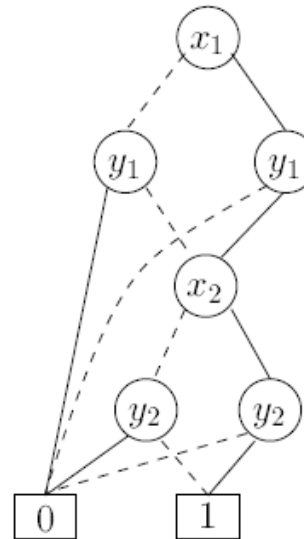
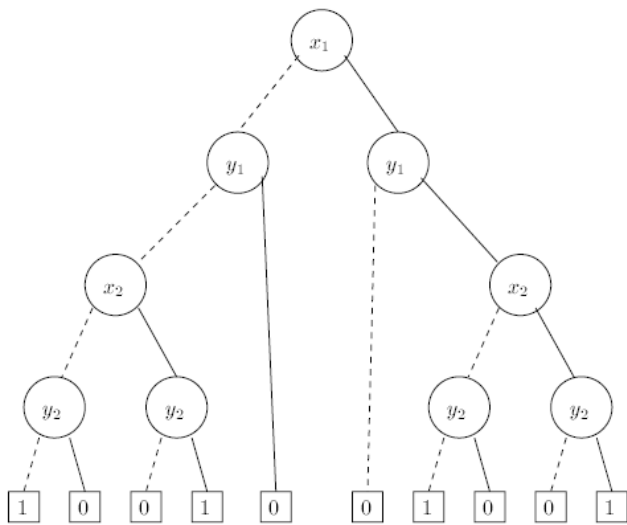
Christian Kästner  
Carnegie Mellon University  
USA

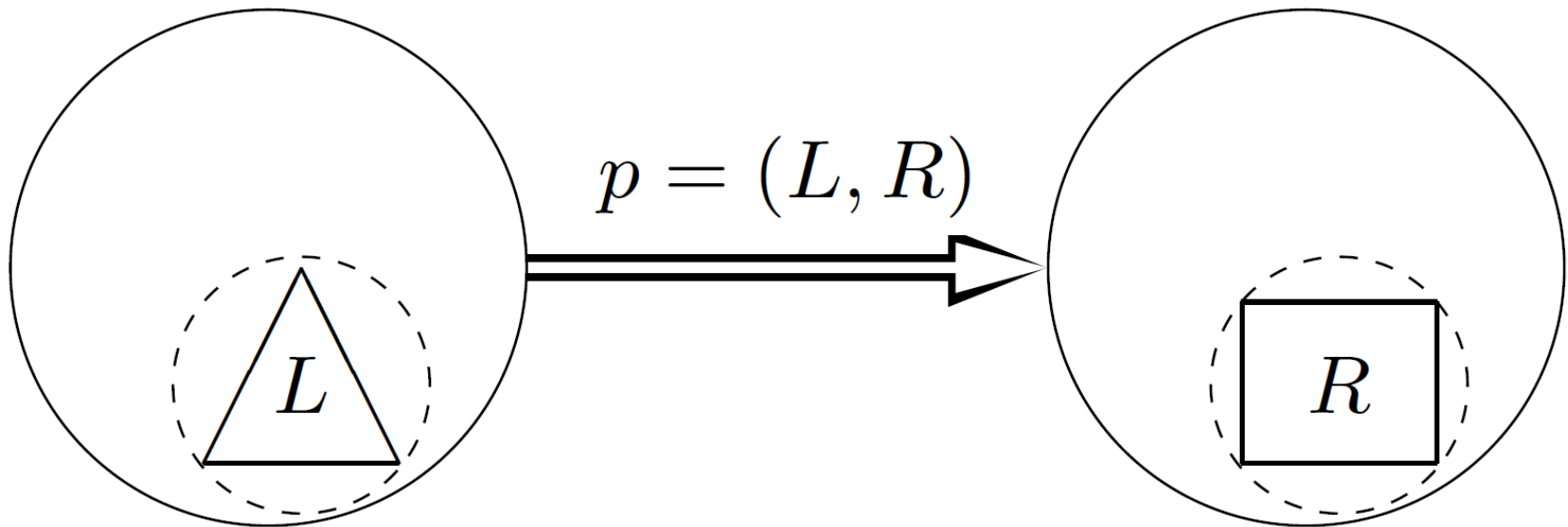
Martin Erwig  
Oregon State University  
USA

Sven Apel  
University of Passau  
Germany

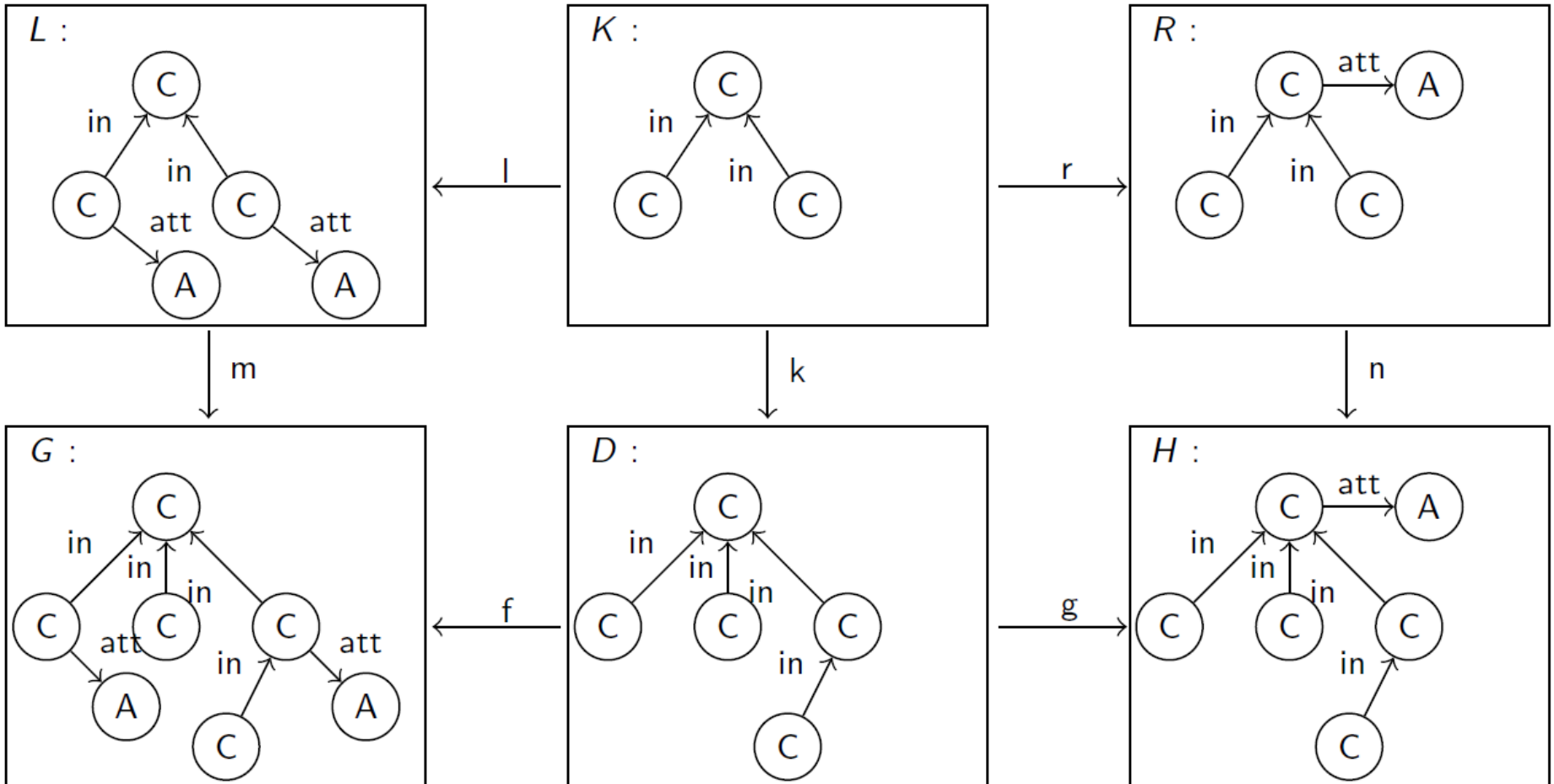
Eric Bodden  
Fraunhofer SIT & TU Darmstadt  
Germany

- $(x_1 \Leftrightarrow y_1) \wedge (x_2 \Leftrightarrow y_2)$

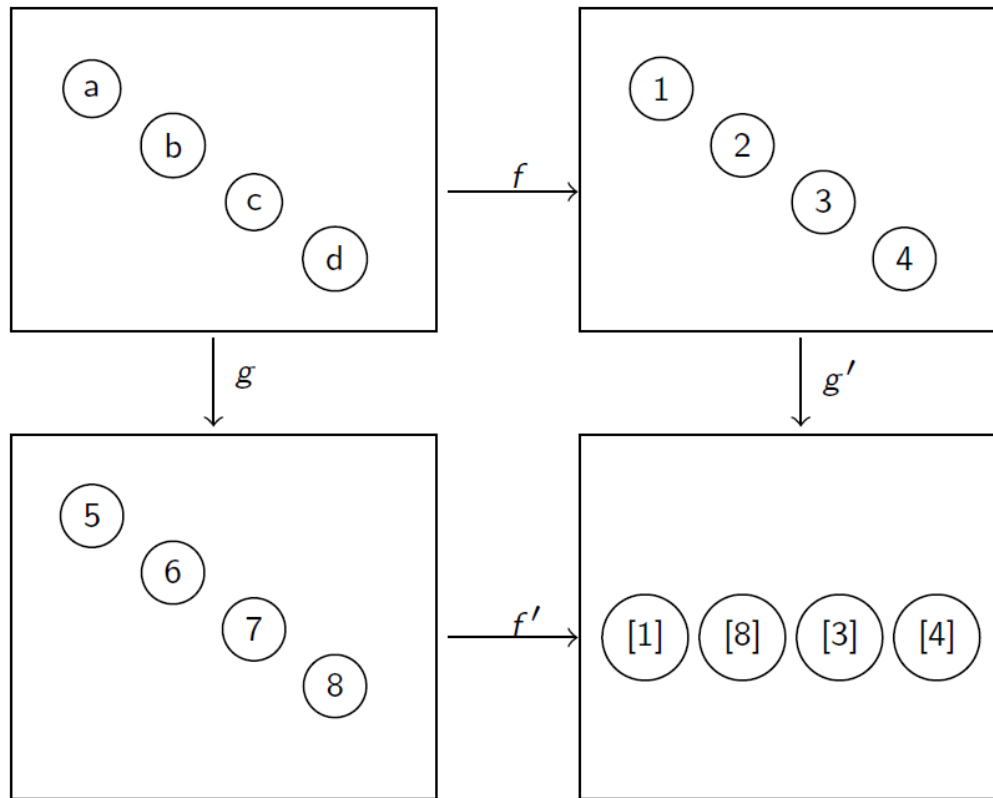




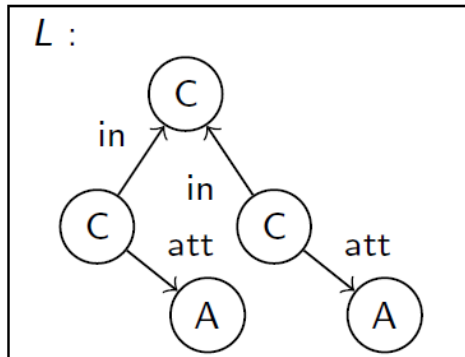
## • Double-Pushout



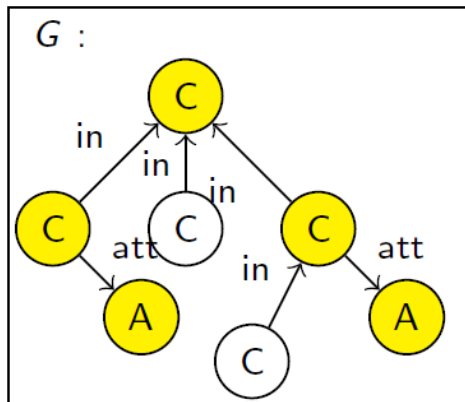
- Pushout, für Mengen



- Wie findet man einen Teilgraphen



m





- **Bis Mittwoch, den 21.04. 12 Uhr:**
  - Wählen Sie drei bevorzugte Themen aus.
  - Senden Sie die Liste der drei Folientitel und ob Sie Bachelor oder Master studieren an [Robert.Mueller@uni-siegen.de](mailto:Robert.Mueller@uni-siegen.de).
  - Wir bemühen uns, alle Themenwünsche zu berücksichtigen.
- Die Folien finden Sie auf der Homepage:  
<https://mbe.informatik.uni-siegen.de/lehre/modellbasierte-entwicklung/>