Static Analysis: Automated Bug Hunting and Beyond

Julian Erhard    Dr. Michael Petter    Michael Schwarz
{julian.erhard, michael.petter, m.schwarz}@tum.de

Chair for Formal Languages, Compiler Construction, Software Construction
Department of Informatics, Technical University of Munich

Winter Term 23/24
Writing programs is hard.
Writing correct programs is very hard.
Static Analysis

- Fully automated
- Can show absence of certain classes of bugs
- Runs directly on the input program
- Abstract Interpretation, Model Checking, ...
Static Analysis

- Fully automated
- Can show absence of certain classes of bugs
- Runs directly on the input program
- Abstract Interpretation, Model Checking, ...
Abstract Interpretation

- Widely used both in Academia & Industry
- Can scale to huge industry-scale codebases
- The technique covered in Program Optimization Course (IN2053)
- Analysis of multi-threaded, real-world C
- Efficient solvers for computation of fixpoints
- https://goblint.in.tum.de
Example

**Figure:** VS Code with the **GobPie** extension, showing warnings found by **Goblint**.
Relational Domains are a key ingredient to proving properties such as freedom of buffer-overruns

Enhance Goblint with new relational domains, such as

- all valid equalities of the form $x = a \cdot y + c$ for constants $c, a \in \mathbb{Z}$
```c
dataX_t x_arr[100];
dataY_t y_arr[100];
dataX_t *x_ptr;
dataY_t *y_ptr;

int main()
{
    int i;
    x_ptr = &x_arr[0];
    y_ptr = &y_arr[0];

    for(i=0; i<100; i++)
    {
        access();
        x_ptr++;
        y_ptr++;
    }
}
```

**Figure:** Example program from Flexeder et al.
Format

- Course will take place throughout the semester
- Teams of 2-5 students
- At latest 1 month before end of term: PR on GitHub
- Last week of lectures: Final presentation
  - Attendance & Active Participation mandatory(!)
Create a Kanban board on Github, and give us access
You should meet 2-3 times a week to discuss progress, blockers, ...
Every 2 weeks (end of "sprint"): Meeting with us in person to demonstrate results, get feedback, clarify next steps
Benefits

- Prevent the next starship from exploding (maybe)
- Deepen your understanding of
  - The Semantics of C and typical programming errors
  - Static Analysis by Abstract Interpretation
- Level up your functional programming skills
- Become connected to the research we do day-to-day
Requirements

- Program Optimization Course helpful (IN2053)
- Knowledge of a functional programming language (we use OCaml)
- Be in your Master’s (Advanced Bachelor’s students welcome)
Questions?
Further Reading