

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, ...

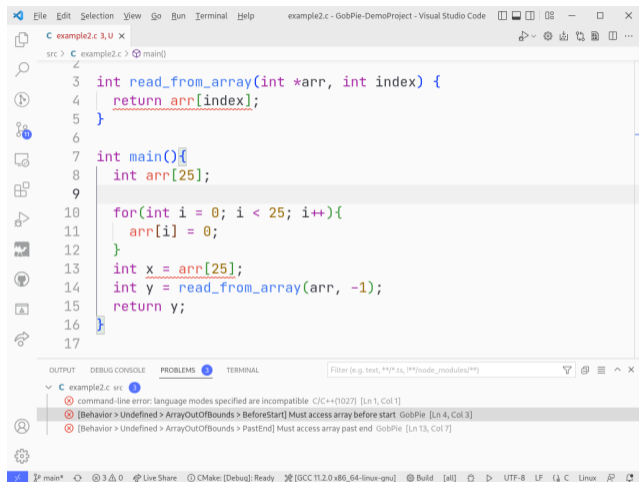
Abstract Interpretation

- ▶ Widely used both in Academia & Industry
- ▶ Can scale to huge industry-scale codebases
- ▶ The technique covered in Program Optimization Course (IN2053)

GOBLINT

- ▶ Analysis of multi-threaded, real-world C
- ▶ Efficient solvers for computation of fixpoints
- ▶ `https://goblint.in.tum.de`

Example



```
File Edit Selection View Go Run Terminal Help example2.c - GobPie-DemoProject - Visual Studio Code
C example2.c 3,U x
src > C example2.c > main()
3 int read_from_array(int *arr, int index) {
4     return arr[index];
5 }
6
7 int main(){
8     int arr[25];
9
10    for(int i = 0; i < 25; i++){
11        arr[i] = 0;
12    }
13    int x = arr[25];
14    int y = read_from_array(arr, -1);
15    return y;
16 }
17
```

OUTPUT DEBUG CONSOLE PROBLEMS TERMINAL

Filter (e.g. text, **/*.ts, !**/node_modules/**)

- command-line error: language modes specified are incompatible C/C++[1027] [Ln 1, Col 1]
- [Behavior > Undefined > ArrayOutOfBounds > BeforeStart] Must access array before start GobPie [Ln 4, Col 3]
- [Behavior > Undefined > ArrayOutOfBounds > PastEnd] Must access array past end GobPie [Ln 13, Col 7]

main* 3.0 Live Share CMake: [Debug] Ready [GCC 11.2.0 x86_64-linux-gnu] Build [all] UTF-8 LF C Linux

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
 - ▶ Flexeder et al. "Fast interprocedural linear two-variable equalities", TOPLAS 2012.
 - ▶ all valid equalities of the form $x = a \cdot y + c$ for constants $c, a \in \mathbb{Z}$

Example

```
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++;
  }
}
```

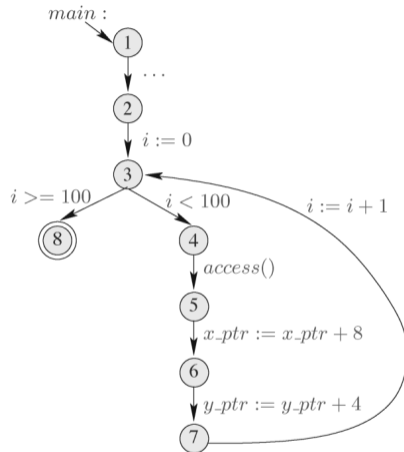


Figure: Example program from Flexeder et al.

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(!)

Lightweight Agile

- ▶ 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



The goblint developers.

Goblint documentation.

URL: <https://goblint.readthedocs.io/en/latest/>.