Karlsruhe Institute of Technology

The Bounded Model Checker LLBMC

Stephan Falke, **Florian Merz, and Carsten Sinz**

INSTITUTE FOR THEORETICAL COMPUTER SCIENCE (ITI)

```
struct list_node {
    int data;
    struct list_node *tail;
};
typedef struct list_node list;

list *reverse(list *l) {
  list *r = l, *p = NULL;
  while (r != NULL) {
    list *q = r;
    r = r->tail;
    q->tail = p;
    p = q;
  }
  return p;
}
```

Uses

- Find bugs and (statically) check assertions

Uses

- Find bugs and (statically) check assertions
- Detect vulnerabilities

Uses

- Find bugs and (statically) check assertions
- Detect vulnerabilities
- Check equivalence of implementations

Uses

- Find bugs and (statically) check assertions
- Detect vulnerabilities
- Check equivalence of implementations
- Abstract (parametric) testing

Uses

- Find bugs and (statically) check assertions
- Detect vulnerabilities
- Check equivalence of implementations
- Abstract (parametric) testing
- Successfully participate in SV-COMP

Goals

- Improve the quality of software

Goals

- Improve the quality of software
- Reduce the time and effort for testing

Goals

- Improve the quality of software
- Reduce the time and effort for testing
- No “false positives”

Goals

- Improve the quality of software
- Reduce the time and effort for testing
- No “false positives”
- Support all features of C

Goals

- Improve the quality of software
- Reduce the time and effort for testing
- No “false positives”
- Support all features of C
- Support most features of C++

LLBMC – Low-Level Bounded Model Checking

Karlsruhe Institute of Technology

- Method: (software) bounded model checking
  - Consider only bounded number of loop iterations
  - Consider only bounded function call depth
  - Use an efficient SMT-solver

LLBMC – Low-Level Bounded Model Checking

Karlsruhe Institute of Technology

- Method: (software) bounded model checking
  - Consider only bounded number of loop iterations
  - Consider only bounded function call depth
  - Use an efficient SMT-solver
- Target: (embedded) C/C++ code

LLBMC – Low-Level Bounded Model Checking

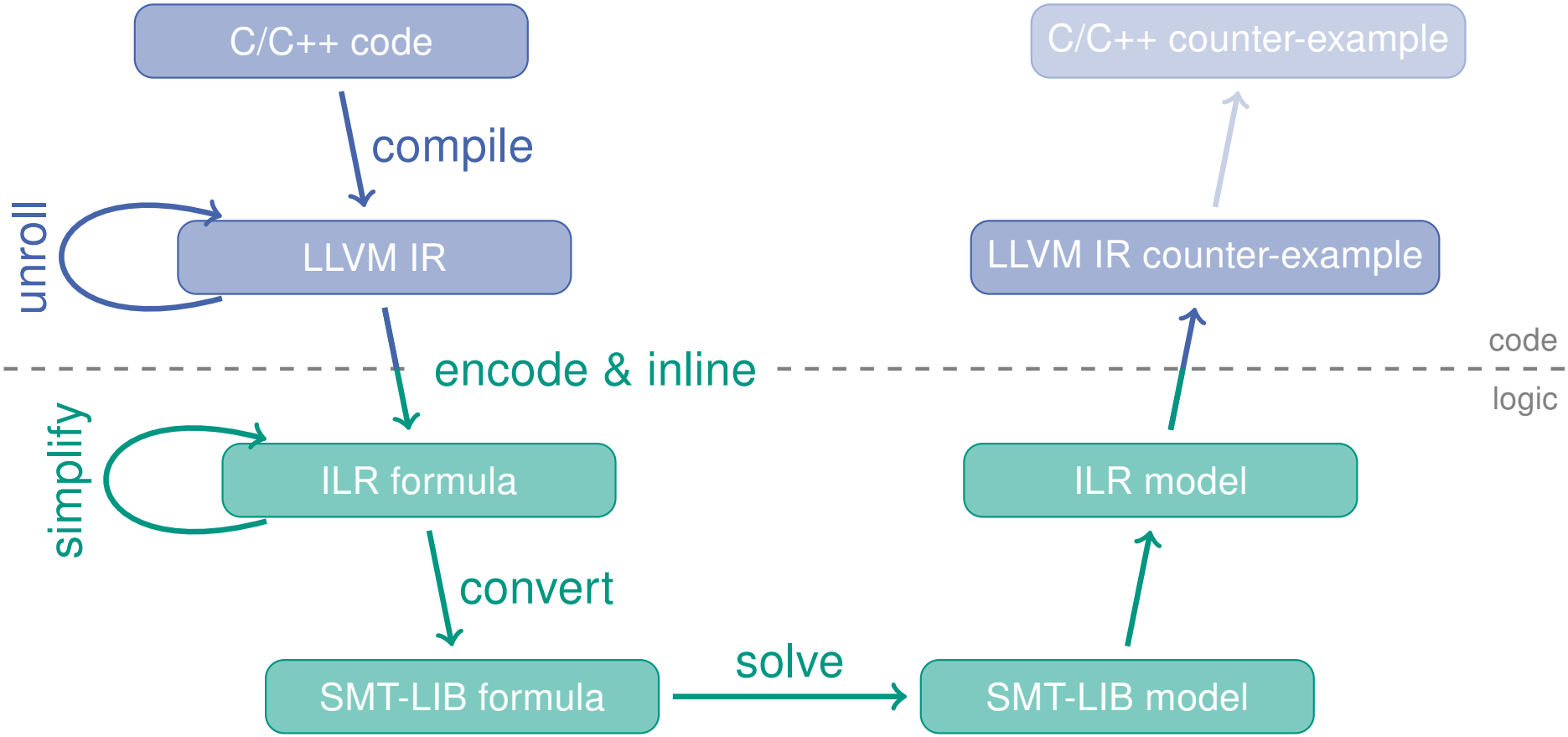
Karlsruhe Institute of Technology

- Method: (software) bounded model checking
  - Consider only bounded number of loop iterations
  - Consider only bounded function call depth
  - Use an efficient SMT-solver
- Target: (embedded) C/C++ code
- Key feature: bit-precision

Built-In Checks

- Integer overflow and underflow
- Division by zero
- Invalid pointer dereference
- Invalid `malloc`
- Invalid `free`
- Stack overflow
- Memory leak
- Invalid bitshift
- User-provided properties (`assume` / `assert`)

LLBMC's Approach



Participation in SV-COMP

- International Competition on Software Verification 2012 and 2013

Participation in SV-COMP

- International Competition on Software Verification 2012 and 2013
- C programs ranging from 13 LOC to 182 kLOC

Participation in SV-COMP

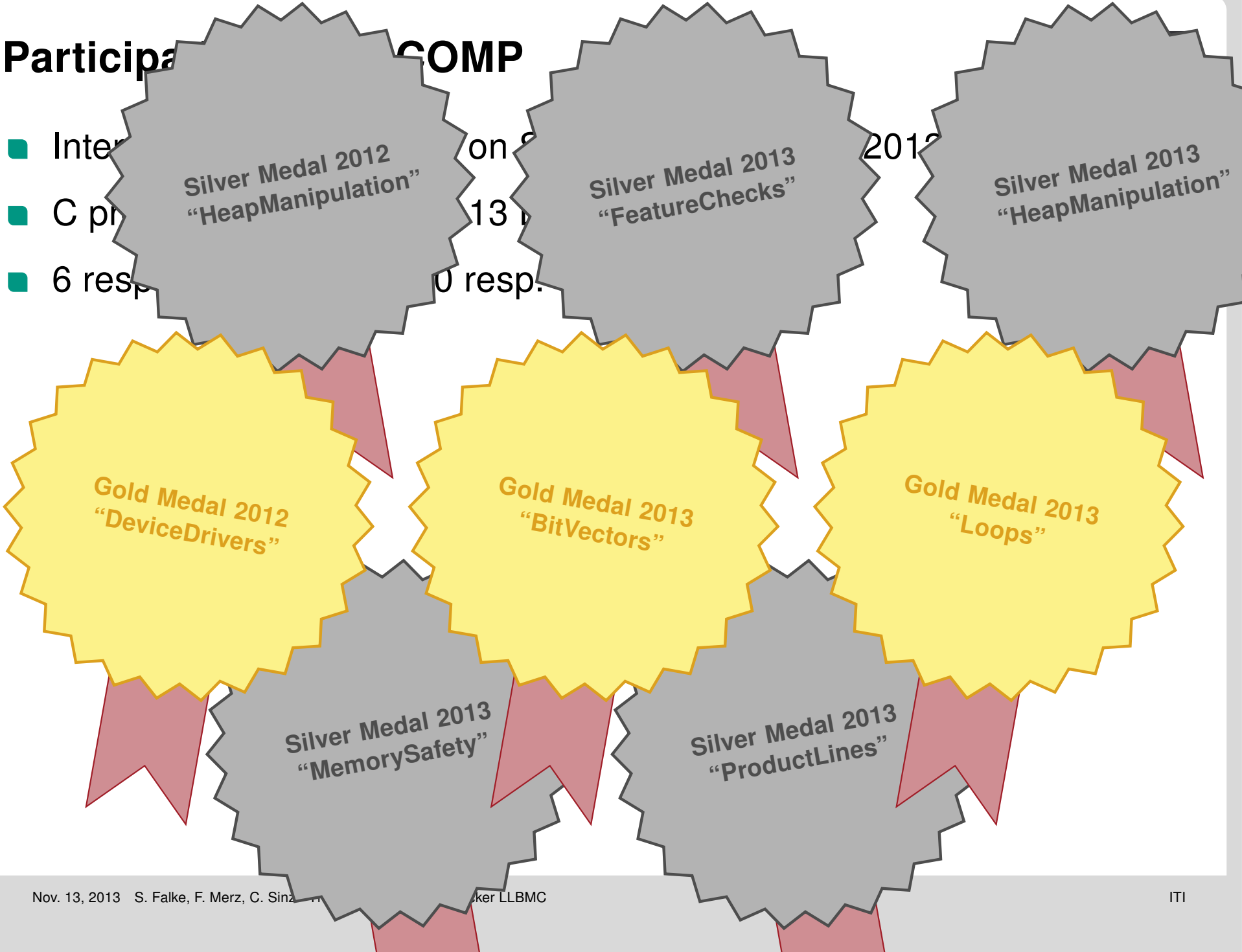
- International Competition on Software Verification 2012 and 2013
- C programs ranging from 13 LOC to 182 kLOC
- 6 resp. 10 categories, 10 resp. 11 participants

Participation in SV-COMP

- International Competition on Software Verification 2012 and 2013
- C programs ranging from 13 LOC to 182 kLOC
- 6 resp. 10 categories, 10 resp. 11 participants

Participa COMP

- Inte on 2012
- C pr 13
- 6 resp 0 resp.



Karlsruhe Institute of Technology

http://llbmc.org