Title: Generating Preconditions for a Modular Analysis based on Bounded Model Checking

Topic:

The autonomous cars of the future will have approx. 300 MLoC and are thus too large for state-of-the-art solvers to be analysed. A promising approach is to partition the program into smaller *modules* and analyse them separately. Looking at the entry point of such a module, all input values are unknown and must be abstracted, which leads to error warnings where really nothing is wrong (false positive).

An approach to analyse such modules is called bounded model checking (BMC). The general idea is to encode paths of a transition system up to a certain bound. After the unrolling and encoding of the program, a formula that represents the negation of a desired property is added, and the formula is solved with an SMT or SAT-solver. If the solver finds a model for the formula, the approach has found an error and the model can be used as a counterexample.

Based on these counterexamples, we want to automatically create preconditions, limiting the input of the module to such values for which the error does not occur.

Simple Example:

```
// 4) (speed != -5 || dist <= 0)
int neededSpace(int speed, int dist){
    int space = 0;
    speed = speed + 5;        // 3) (speed != 0-5 || dist <= 0)
    if(dist > 0){             // 2) (speed != 0 || dist <= 0)
        space = dist/speed; // 1) (speed != 0)
    }
    return space;
}
```

Possible Task Description:

- Analyse information from SAT module and trace from bounded model checker for a limited number of error types (division by zero, arithmetic overflow)
- Define what kind of preconditions can be generated for these errors, also exact ($x=3$) vs. abstraction ($x<10$)
- Generate preconditions through evaluating values along the trace, probably with SAT solving
- Create a simplifier for the generated conditions (simple example: $x \leq 10-5 \rightarrow x \leq 5$)

Names of Supervisors:

Marko Kleine Büning, marko.kleinebuening@kit.edu, 50.34, Room 017
Prof. Carsten Sinz, carsten.sinz@kit.edu, 50.34, Room 028

Possible Number of Participants: 1 to 2 students (if a group of more than two students wants to work on this project, the scope can be lifted with our consultation)