Extending DIVINE with Symbolic Verification using SMT

Henrich Lauko, Vladimír Štill, Petr Ročkai and Jiří Barnat

Masaryk University

April 5, 2019
- **Explicit-state** model checker for C/C++
- Based on the **LLVM** toolchain
- Support of control-flow non-determinism – **parallelism**
- Reachability, LTL, assertions, memory safety, deadlocks
Let program do the symbolic computation.

Concrete program:

```c
1  int a = __nondet();
2  int b = factorial(7);
3  int c = a + b;
4  if (a == c) {
5     ...
6 }
```

Symbolic program:

```c
1  sym_int a = __sym_val();
2  int b = factorial(7);
3  sym_int c = s_add(a,b);
4  sym_bool d = s_eq(a,c);
5  if (nondet_bool()) {
6     assume(d);
7     ...
8 }
```
Extending DIVINE with Symbolic Verification

No need to complicate the verification core.

https://divine.fi.muni.cz