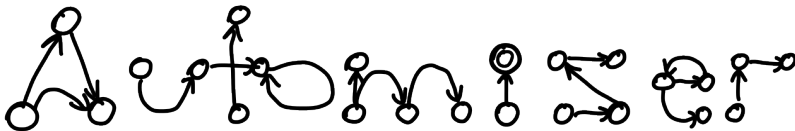


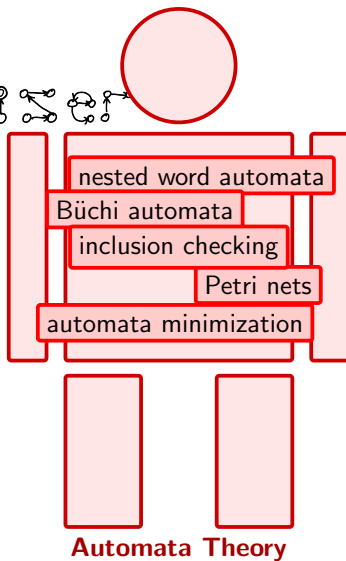
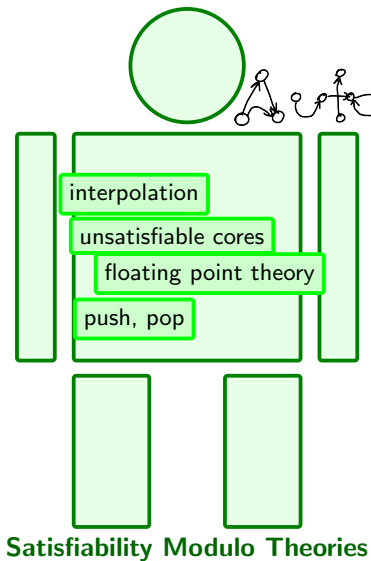
ULTIMATE



automata-based software verification

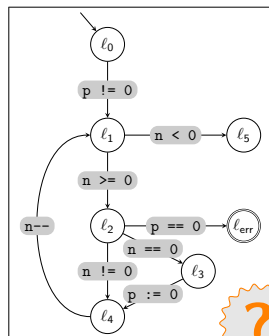
for **non-reachability**, **memory safety**, **termination**, **overflows**

2023 competition team: [Matthias Heizmann](#), [Max Barth](#),
[Daniel Dietsch](#), [Leonard Fichtner](#), [Jochen Hoenicke](#),
[Dominik Klumpp](#), [Mehdi Naouar](#), [Tanja Schindler](#),
[Frank Schüssele](#), [Andreas Podelski](#)

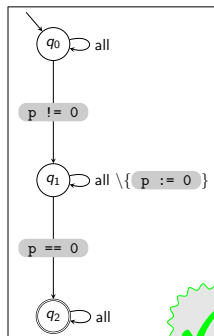


Automata

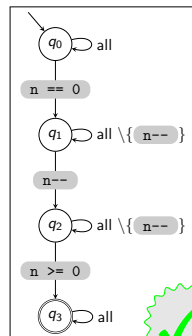
Automata-theoretic decomposition of verification tasks



program \mathcal{P}



program \mathcal{P}_1



program \mathcal{P}_2



$$L(\mathcal{P}) \subseteq L(\mathcal{P}_1) \cup L(\mathcal{P}_2)$$

Uni-Freiburg : SWT - Ultimate — Mozilla Firefox

Uni-Freiburg : SWT - Ultima × +

https://ultimate-pa.org/?ui=int&tool=automizer#

Ultimate Automizer Language: C ▾ Samples ▾ Execute Settings ▾

```
20
21 int main() {
22     int p = 42;
23     int n = __VERIFIER_nondet_int();
24     while ( n >= 0 ) {
25         //@ assert p != 0;
26         if (n == 0) {
27             p = 0;
28         }
29         n--;
30     }
31     return 0;
32 }
33
```

i assertion always holds

Line	Col	Description
25	12	For all program executions holds that assertion always holds at this location

<https://ultimate-pa.org/?ui=tool&tool=automizer>