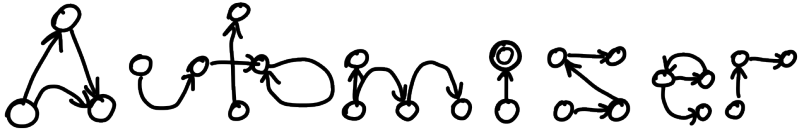


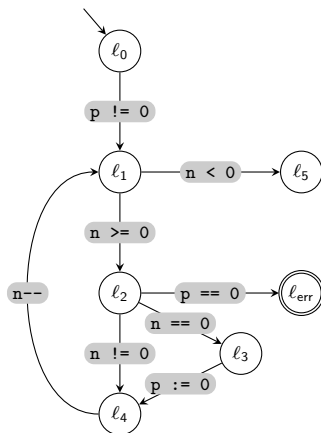
ULTIMATE



with Unsatisfiable Cores

Matthias Heizmann, Jürgen Christ, Daniel Dietsch,
Jochen Hoenicke, Markus Lindenmann, Betim Musa,
Christian Schilling, Stefan Wissert, Andreas Podelski

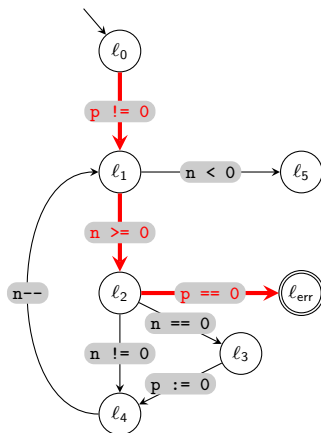
Example



control flow graph of \mathcal{P}

Example

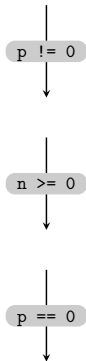
1. pick some error trace



control flow graph of \mathcal{P}

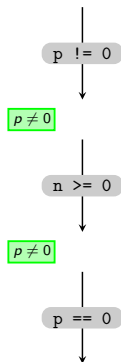
Example

1. pick some error trace
2. is trace feasible?



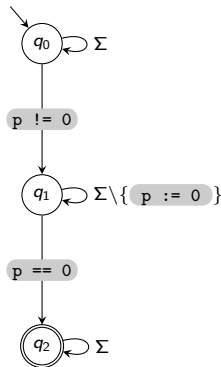
Example

1. pick some error trace
2. is trace feasible?
3. reason for infeasibility?



Example

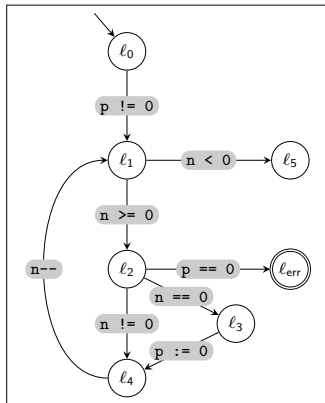
1. pick some error trace
2. is trace feasible?
3. reason for infeasibility?
4. generalize
from single infeasible trace
to set of infeasible traces



Example - second iteration

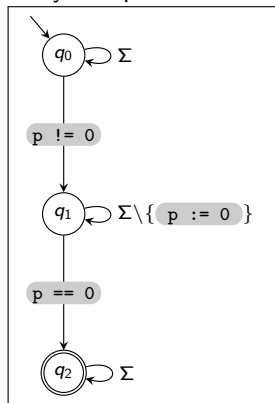
restart verification of \mathcal{P}

but exclude all error traces whose infeasibility was proven



program \mathcal{P}

set of error traces

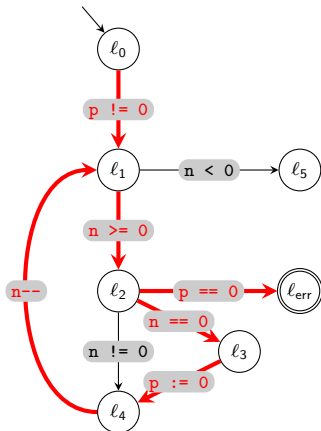


automaton \mathcal{A}_1

set of infeasible traces

Example - second iteration

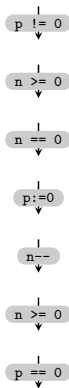
1. pick some error trace



control flow graph of \mathcal{P}

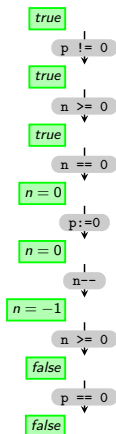
Example - second iteration

1. pick some error trace
2. is trace feasible?



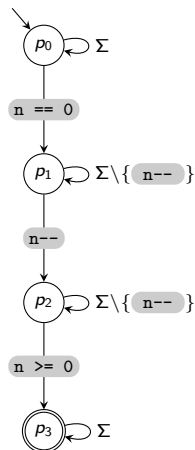
Example - second iteration

1. pick some error trace
2. is trace feasible?
3. reason for infeasibility?

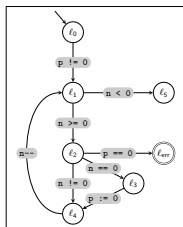


Example - second iteration

1. pick some error trace
2. is trace feasible?
3. reason for infeasibility?
4. generalize
from single infeasible trace
to set of infeasible traces

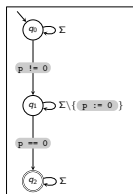


Example - each error trace is infeasible



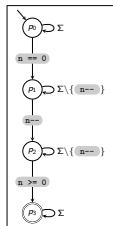
program \mathcal{P}
set of error traces

\cup



automaton \mathcal{A}_1
set of infeasible
traces

\cup



automaton \mathcal{A}_2
set of infeasible
traces

interprocedural analysis and recursive programs

- ▶ nested word automata

*H., Hoenicke, Podelski: **Nested Interpolants** (POPL 2010)*

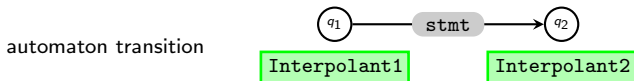
interprocedural analysis and recursive programs

- ▶ nested word automata

H., Hoenicke, Podelski: Nested Interpolants (POPL 2010)

generalization from single infeasible trace to set of infeasible traces

Hoare triple { Interpolant1 } stmt { Interpolant2 }



H., Hoenicke, Podelski: Software Model Checking for People who Love Automata (CAV 2013)

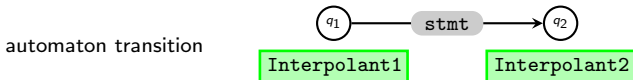
interprocedural analysis and recursive programs

- ▶ nested word automata

H., Hoenicke, Podelski: Nested Interpolants (POPL 2010)

generalization from single infeasible trace to set of infeasible traces

Hoare triple { Interpolant1 } stmt { Interpolant2 }



H., Hoenicke, Podelski: Software Model Checking for People who Love Automata (CAV 2013)

termination analysis

- ▶ Büchi automata

Clark



Abstract
Introduction
Methods
Results
Discussion
Conclusion

ocres

ULTIMATE SUBSTITUTION

Center
aires



ULTIMATE WEB-INTERFACE

Task:

Verify C

Sample:

NullpointerAfterLastiterati...

Tool:

Trace abstraction

Trace Abstraction

toolchain

SETTINGS

```
i 1  int main() {
2     int p, n;
3     p = 1;
i 4  while ( n>=0 ) {
i 5     //@ assert p != 0;
6     if (n == 0) {
7         p = 0;
8     }
9     n--;
10  }
11  return 0;
12 }
13 }
```

EXECUTE



[Show editor fullscreen](#)

Browse...

	Line	Ultimate Result
	5	assertion always holds
	0	Ultimate Automizer runtime statistics Ultimate Automizer took 121ms
	1 - 12	Procedure Contract for main true
	4 - 10	Loop Invariant ((n<=(-1)) !(0<=(p+(-1))))

<http://ultimate.informatik.uni-freiburg.de/automizer>