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 $P$
1. pick some error trace

2. is trace feasible?

\[
\begin{align*}
p \neq 0 \\
n \geq 0 \\
p = 0
\end{align*}
\]
Example

1. pick some error trace

2. is trace feasible?

3. reason for infeasibility?
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

\[
\ell_0 \quad p \neq 0 \quad q_0 \quad \Sigma
\]

\[
\ell_1 \quad n < 0 \quad \ell_2 \quad n \geq 0
\]

\[
\ell_2 \quad p = 0 \quad \ell_3 \quad n \neq 0
\]

\[
\ell_3 \quad p := 0 \quad \ell_4 \quad n := 0
\]

\[
\ell_4 \quad n := 0 \quad \ell_5
\]

program \( \mathcal{P} \)

set of error traces

automaton \( \mathcal{A}_1 \)

set of infeasible traces
1. pick some error trace
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 $P$
set of error traces

$\subseteq$

automaton $A_1$
set of infeasible traces

$\cup$

automaton $A_2$
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

\[ H., Hoenicke, Podelski: \textbf{Nested Interpolants} \text{ (POPL 2010)} \]

generalization from single infeasible trace to set of infeasible traces

Hoare triple

\[
\{ \text{Interpolant1} \} \quad \text{stmt} \quad \{ \text{Interpolant2} \}
\]

automaton transition

\[
q_1 \xrightarrow{\text{stmt}} q_2
\]

\[ H., Hoenicke, Podelski: \textbf{Software Model Checking for People who Love Automata} \text{ (CAV 2013)} \]
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

\[
\{ \text{Interpolant1} \} \xrightarrow{\text{stmt}} \{ \text{Interpolant2} \}
\]

automaton transition

\[
q_1 \xrightarrow{\text{stmt}} q_2
\]

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

termination analysis

- Büchi automata
int main() {
    int p, n;
    p = 1;
    while (n >= 0) {
        // assert p != 0;
        if (n == 0) {
            p = 0;
        }
        n--;
    }
    return 0;
}