

THETA: Algorithms Complementing Abstraction Techniques

*Csanád Telbisz, Milán Mondok, Levente Bajczi,
Zsófia Ádám, Márk Somorjai, Balázs Rippl, Dániel
Kovács, Mihály Dobos-Kovács*



**Critical Systems
Research Group**

Program Verification Formalisms and Algorithms

Input
formalism

Analysis
formalisms

Analysis
algorithms

Program Verification Formalisms and Algorithms

Input
formalism

CFA

Analysis
formalisms

Analysis
algorithms

Program Verification Formalisms and Algorithms

Input
formalism

CFA

Analysis
formalisms

LTS
*labelled
transition system*

STS
*symbolic
transition system*

Event graph

Analysis
algorithms

Program Verification Formalisms and Algorithms

Input
formalism

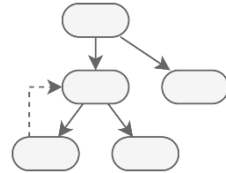
CFA

Analysis
formalisms

LTS
*labelled
transition system*

STS
*symbolic
transition system*

Event graph



Analysis
algorithms

Program Verification Formalisms and Algorithms

Input
formalism

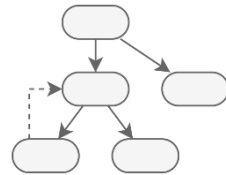
CFA

Analysis
formalisms

LTS
*labelled
transition system*

STS
*symbolic
transition system*

Event graph



Analysis
algorithms

CEGAR

Liveness
CEGAR

Program Verification Formalisms and Algorithms

Input
formalism

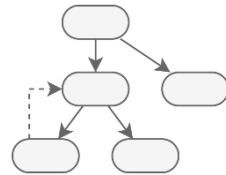
CFA

Analysis
formalisms

LTS
*labelled
transition system*

STS
*symbolic
transition system*

Event graph



Analysis
algorithms

CEGAR

original Theta
approach

Liveness
CEGAR

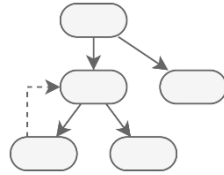
Program Verification Formalisms and Algorithms

Input
formalism

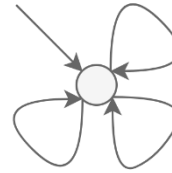
CFA

Analysis
formalisms

LTS
*labelled
transition system*



STS
*symbolic
transition system*



Event graph

Analysis
algorithms

CEGAR

original Theta
approach

Liveness
CEGAR

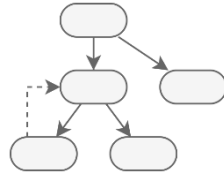
Program Verification Formalisms and Algorithms

Input
formalism

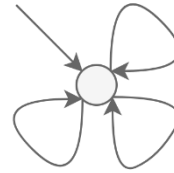
CFA

Analysis
formalisms

LTS
*labelled
transition system*



STS
*symbolic
transition system*



Event graph

Analysis
algorithms

CEGAR

BMC

original Theta
approach

Liveness
CEGAR

K-induction

IMC

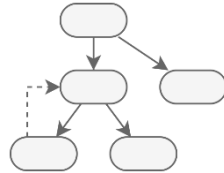
Program Verification Formalisms and Algorithms

Input
formalism

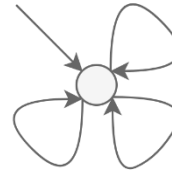
CFA

Analysis
formalisms

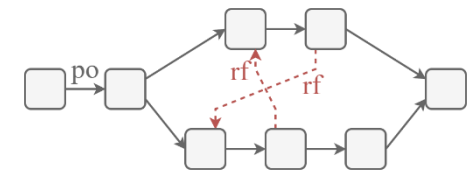
LTS
*labelled
transition system*



STS
*symbolic
transition system*



Event graph



Analysis
algorithms

CEGAR

BMC

Liveness
CEGAR

K-induction

IMC

original Theta
approach

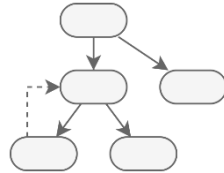
Program Verification Formalisms and Algorithms

Input
formalism

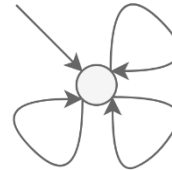
CFA

Analysis
formalisms

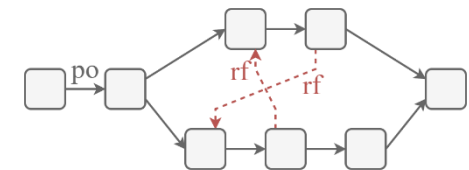
LTS
*labelled
transition system*



STS
*symbolic
transition system*



Event graph



Analysis
algorithms

CEGAR

BMC

Ordering
consistency
checker

original Theta
approach

Liveness
CEGAR

K-induction

IMC

Supported Properties

2026	Native support (dedicated algorithm)	Support via specification transformation
Reachability	✓	–
Data race	✓	✓
Termination	✓	✓
Memory safety	✗	✓
Overflow	✗	(✓)
<i>Validation</i>	✗	✓

Supported Properties

2024	Native support (dedicated algorithm)	Support via specification transformation
Reachability	✓	–
Data race	✓	✗
Termination	✗	✗
Memory safety	✗	(✓)
Overflow	✗	(✓)
<i>Validation</i>	✗	✗

Supported Properties

2026	Native support (dedicated algorithm)	Support via specification transformation
Reachability	✓	–
Data race	✓	✓
Termination	✓	✓
Memory safety	✗	✓
Overflow	✗	(✓)
<i>Validation</i>	✗	✓

Theta UI

<https://theta.mit.bme.hu/>

Theta

Verification Demo

C

Examples

Property: Unreach Call

property does not hold

```
1 // Taken from https://gitlab.com/sosy-lab/benchmarking/sv-benchmarks/-/blob/main/c/locks/test_locks.c
2
3 extern void abort(void);
4 extern void __assert_fail(const char *, const char *, unsigned int, const char *) __attribute__((noreturn));
5 void reach_error() { __assert_fail("0", "test_locks_14-2.c", 3, "reach_error"); }
6
7 extern int __VERIFIER_nondet_int();
8 int main()
9 {
10     int p1 = __VERIFIER_nondet_int( [0] function_return: \result == 0 ); // condition variable
11     int lk1; // lock variable
12
13     int p2 = __VERIFIER_nondet_int( [1] function_return: \result == 0 ); // condition variable
14     int lk2; // lock variable
15
16     int p3 = __VERIFIER_nondet_int( [2] function_return: \result == 1 ); // condition variable
17     int lk3; // lock variable
18
19     int p4 = __VERIFIER_nondet_int( [3] function_return: \result == 0 ); // condition variable
20     int lk4; // lock variable
21
22     int p5 = __VERIFIER_nondet_int( [4] function_return: \result == 0 ); // condition variable
```

v6.27.14 Custom Run

--input %in --property %prop --backend PORTFOLIO

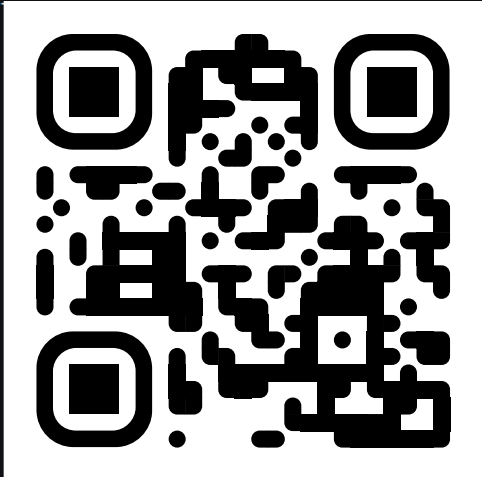
Output

Generated Files

Debug

java -jar /home/cloud/theta-webui/backend/data/theta-cache/v6.27.14/theta-xcfa-cli.jar --input /home/cloud/theta-webui/backend/tmp/run-1776012327575-jtj82s/input.c --property unreach-call.prp --backend PORTFOLIO

Current configuration: KIND-Z3:new-true
subprocess: (Property unreach-call)
subprocess: (SafetyResult Unsafe Trace length: 18)
Config KIND-Z3:new-true succeeded in 14370 ms



Ln 9, Col 2

Theta licensed under Apache-2.0. SMT solver may have different licensing. Check before using.

Theta UI

<https://theta.mit.bme.hu/>

Theta

Verification Demo

C

Examples

Property: Unreach Call

property does not hold

```
1 // Taken from https://gitlab.com/sosy-lab/benchmarking/sv-benchmarks/-/blob/main/c/locks/test_lo
2
3 extern void abort(void);
4 extern void __assert_fail(const char *, const char *, int, const char *) __attribute__((noreturn));
5 void reach_error() { __assert_fail("0", "test_lo
6
7 extern int __VERIFIER_nondet_int();
8 int main()
9 {
10     int p1 = __VERIFIER_nondet_int( [0] function_return: \result == 0 ); // condition variable
11     int lk1; // lock variable
12
13     int p2 = __VERIFIER_nondet_int( [1] function_return: \result == 0 ); // condition variable
14     int lk2; // lock variable
15
16     int p3 = __VERIFIER_nondet_int( [2] function_return: \result == 1 ); // condition variable
17     int lk3; // lock variable
18
19     int p4 = __VERIFIER_nondet_int( [3] function_return: \result == 0 ); // condition variable
20     int lk4; // lock variable
21
22     int p5 = __VERIFIER_nondet_int( [4] function_return: \result == 0 ); // condition variable
```

annotated witness info

v6.27.14

Custom

Run

--input %in --property %prop --backend PORTFOLIO

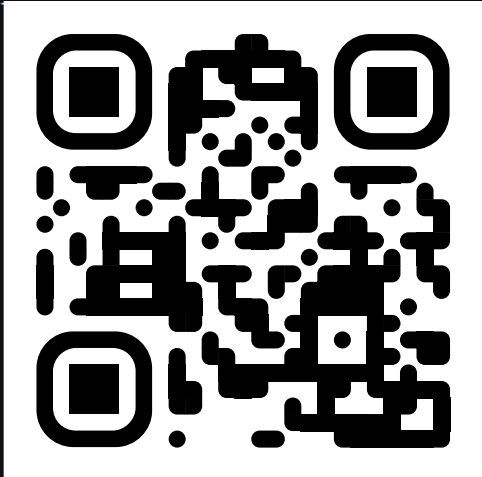
Output

Generated Files

Debug

java -jar /home/cloud/theta-webui/backend/data/theta-cache/v6.27.14/theta-xcfa-cli.jar --input /home/cloud/theta-webui/backend/tmp/run-1776012327575-jtj82s/input.c --property unreachable.prp --backend PORTFOLIO

Current configuration: KIND-Z3:new-true
subprocess: (Property unreachable-call)
subprocess: (SafetyResult Unsafe Trace length: 18)
Config KIND-Z3:new-true succeeded in 14370 ms



Ln 9, Col 2

Theta licensed under Apache-2.0. SMT solver may have different licensing. Check before using.