Verified Rust Monitors for Lola Specifications

Bernd Finkbeiner, Stefan Oswald,
Noemi Passing, *Maximilian Schwenger*

CISPA
HELMHOLTZ CENTER FOR
INFORMATION SECURITY

An Age-Old Question

QUIS CUSTODIET IPSOS CUSTODES?

~ Juvenal (100-200ad)

Quis Custodiet Ipsos Custodes?

Cannot be trusted!

observes

Monitor

Quis Custodiet Ipsos Custodes?

Cannot be trusted!

observes

Monitor

Why trust the monitor?

Quis Custodiet Ipsos Custodes?

Cannot be trusted!

observes

Our Goal:
Verify the Monitor!

Monitor

Why trust the monitor?

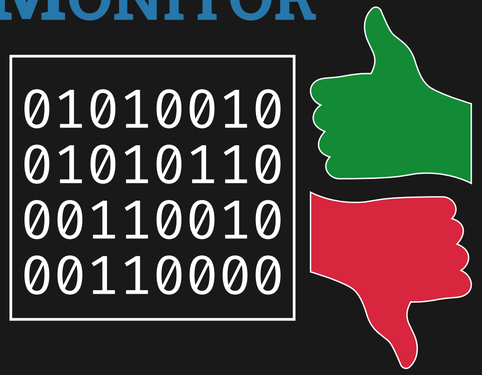
Quis Custodiet Ipsos Custodes?

observes

Our Goal:
Verify the Monitor!

Lola Specification

Monitor



Quis Custodiet Ipsos Custodes?

observes

Our Goal:
Verify the Monitor!

Lola Specification

Compilation

```
Impl Monitor {

 while let Some(i)
        = get_input() {
  …
 }}
```

Rust Code

Monitor

```
01010010
01010110
00110010
00110000
```

Quis Custodiet Ipsos Custodes?

observes

Our Goal:
Verify the Monitor!

Lola Specification

Compilation

+ Annotation Generation

```
Impl Monitor {
 #[invariant = … ]
 while let Some(i)
        = get_input() {
  …
 }}
```

Rust Code

Monitor

```
01010010
01010110
00110010
00110000
```

Quis Custodiet Ipsos Custodes?

VIPER

verifies

observes

Our Goal:
Verify the Monitor!

Lola Specification

Compilation

+ Annotation Generation

```
Impl Monitor {
 #[invariant = … ]
 while let Some(i)
        = get_input() {
  …
 }}
```

Rust Code

Monitor

```
01010010
01010110
00110010
00110000
```

Quis Custodiet Ipsos Custodes?

VIPER

verifies

observes

Our Goal:
Verify the Monitor!

Lola Specification

Compilation

+ Annotation Generation

```
Impl Monitor {
 #[invariant = … ]
 while let Some(i)
        = get_input() {
  …
 }}
```

Rust Code

Monitor

```
01010010
01010110
00110010
00110000
```

Stream-Based Monitoring with Lola

sensors

$i_{1,1}$	$i_{1,2}$	$i_{1,3}$	$i_{1,4}$	$i_{1,5}$	$i_{1,6}$	$i_{1,7}$	$i_{1,8}$
$i_{2,1}$	$i_{2,2}$	$i_{2,3}$	$i_{2,4}$	$i_{2,5}$	$i_{2,6}$	$i_{2,7}$	$i_{2,8}$
$i_{3,1}$	$i_{3,2}$	$i_{3,3}$	$i_{3,4}$	$i_{3,5}$	$i_{3,6}$	$i_{3,7}$	$i_{3,8}$
$o_{1,1}$	$o_{1,2}$	$o_{1,3}$	$o_{1,4}$	$o_{1,5}$	$o_{1,6}$	$o_{1,7}$	$o_{1,8}$
$o_{2,1}$	$o_{2,2}$	$o_{2,3}$	$o_{2,4}$	$o_{2,5}$	$o_{2,6}$	$o_{2,7}$	$o_{2,8}$
$t_{1,1}$	$t_{1,2}$	$t_{1,3}$	$t_{1,4}$	$t_{1,5}$	$t_{1,6}$	$t_{1,7}$	$t_{1,8}$

```
input i1

input i2

input i3

output o1 := i3 + 3

output o2 := i1 + i2 + o1

trigger o2 > 7
```

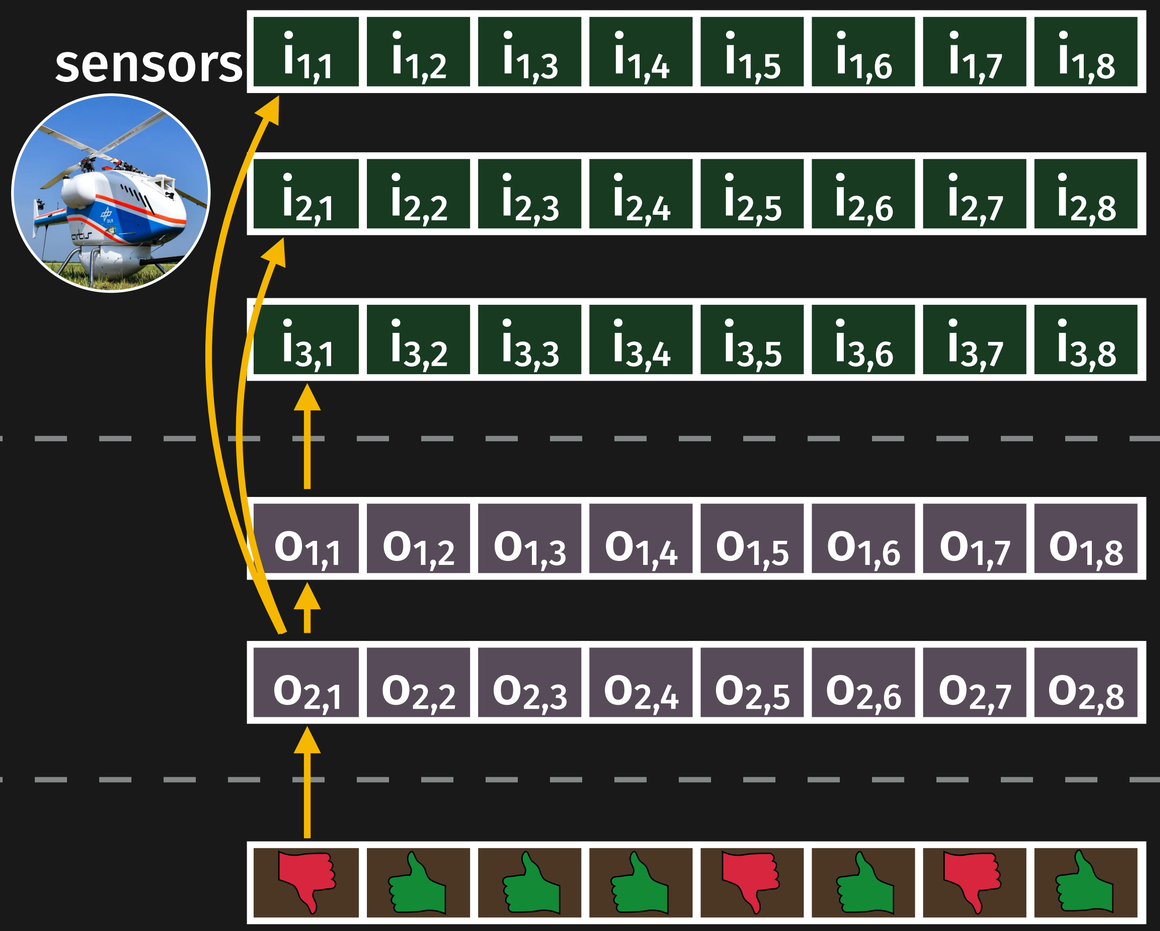
Stream-Based Monitoring with Lola

sensors

$i_{1,1}$	$i_{1,2}$	$i_{1,3}$	$i_{1,4}$	$i_{1,5}$	$i_{1,6}$	$i_{1,7}$	$i_{1,8}$
$i_{2,1}$	$i_{2,2}$	$i_{2,3}$	$i_{2,4}$	$i_{2,5}$	$i_{2,6}$	$i_{2,7}$	$i_{2,8}$
$i_{3,1}$	$i_{3,2}$	$i_{3,3}$	$i_{3,4}$	$i_{3,5}$	$i_{3,6}$	$i_{3,7}$	$i_{3,8}$
$o_{1,1}$	$o_{1,2}$	$o_{1,3}$	$o_{1,4}$	$o_{1,5}$	$o_{1,6}$	$o_{1,7}$	$o_{1,8}$
$o_{2,1}$	$o_{2,2}$	$o_{2,3}$	$o_{2,4}$	$o_{2,5}$	$o_{2,6}$	$o_{2,7}$	$o_{2,8}$
$t_{1,1}$	$t_{1,2}$	$t_{1,3}$	$t_{1,4}$	$t_{1,5}$	$t_{1,6}$	$t_{1,7}$	$t_{1,8}$

```
input i1

input i2

input i3

output o1 := i3 + 3

output o2 := i1 + i2 + o1

trigger o2 > 7
```

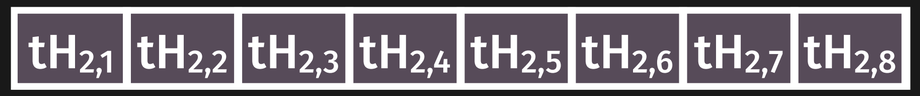
Stream-Based Monitoring with Lola




```
input i1

input i2

input i3

output o1 := i3 + 3

output o2 := i1 + i2 + o1

trigger o2 > 7
```

Stream-Based Monitoring with Lola

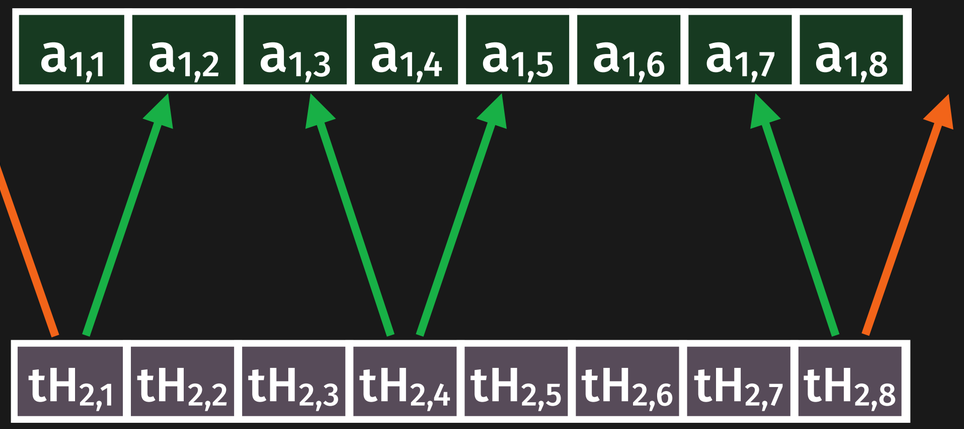
sensors



```
input alt

output tooHigh :=
  alt.offset(by: -1, dft: 0) > 500
  ∧ alt > 500
  ∧ alt.offset(by: +1, dft: 0) > 500

trigger tooHigh
```

Stream-Based Monitoring with Lola

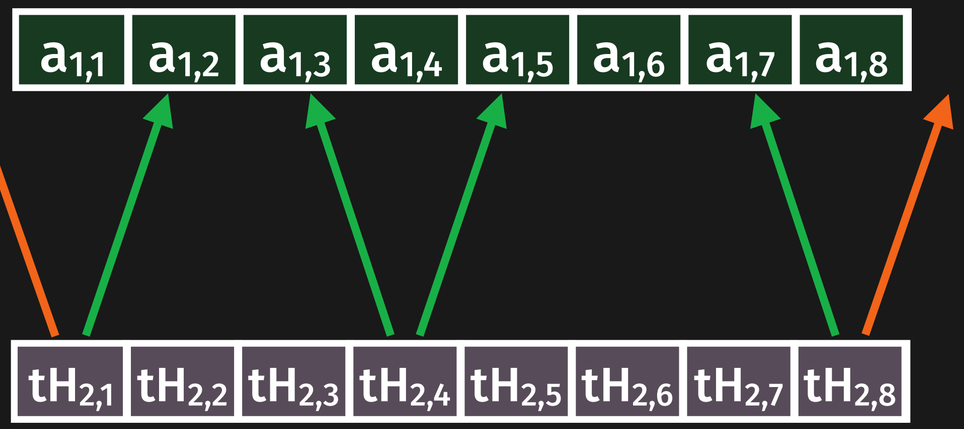
sensors



```
input alt

output tooHigh :=
  alt.offset(by: -1, dft: 0) > 500
  ∧ alt > 500
  ∧ alt.offset(by: +1, dft: 0) > 500

trigger tooHigh
```

Stream-Based Monitoring with Lola

sensors

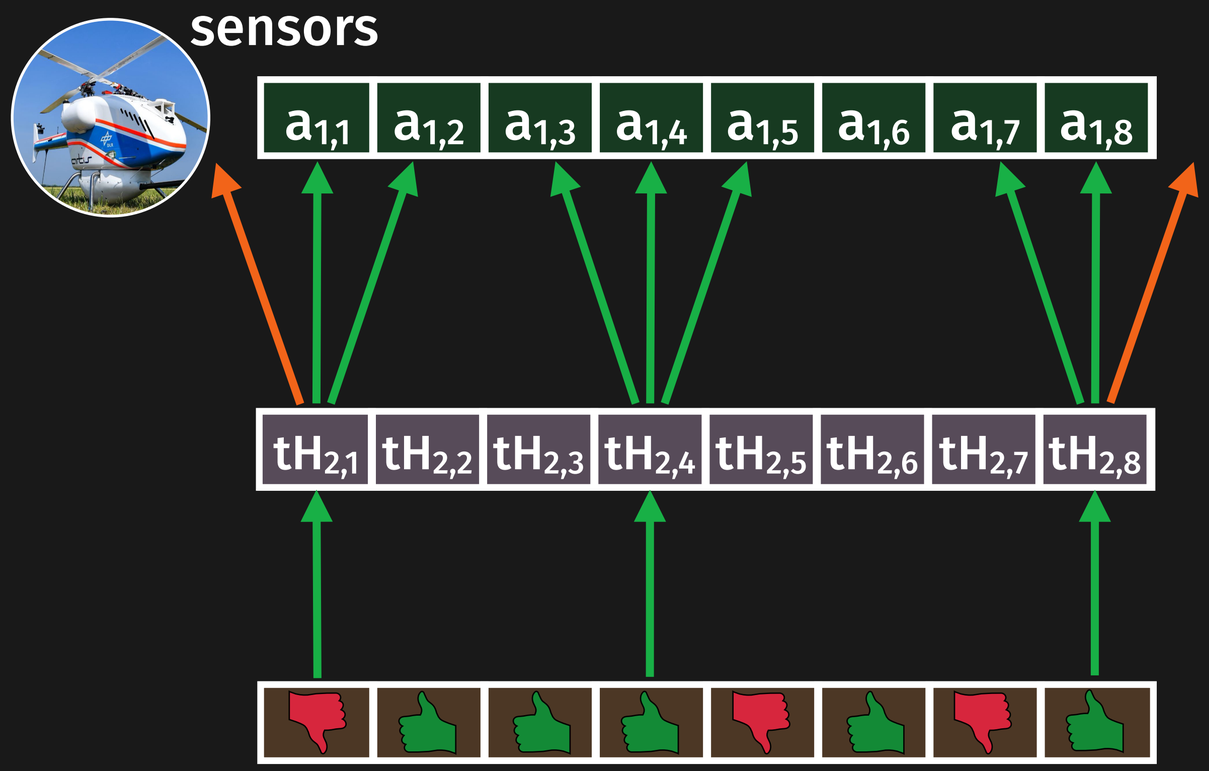


```
input alt

output tooHigh :=
  alt.offset(by: -1, dft: 0) > 500
  ∧ alt > 500
  ∧ alt.offset(by: +1, dft: 0) > 500

trigger tooHigh
```

fails @ t=1

fails @ t=|σ|

Stream-Based Monitoring with Lola



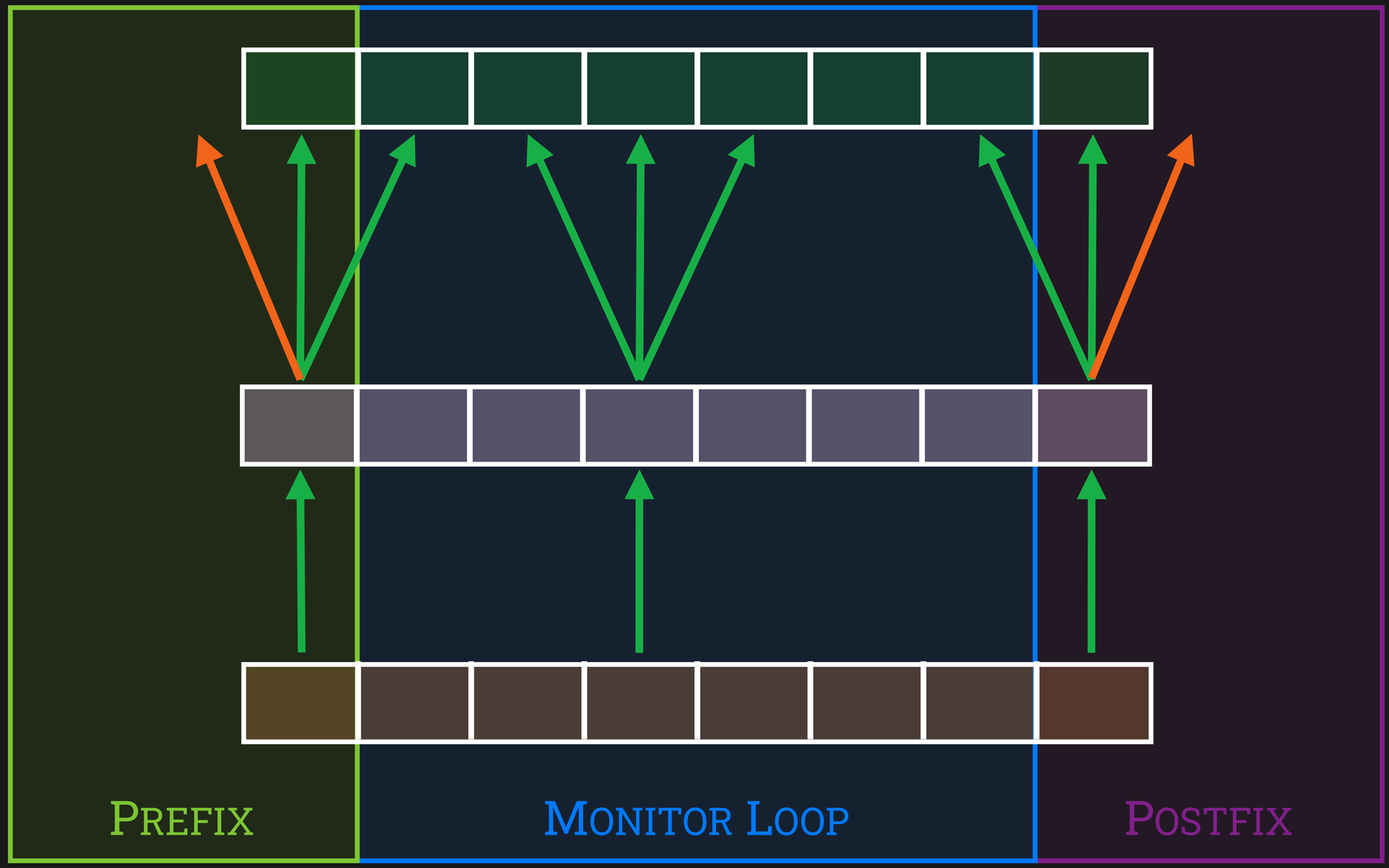

```
input alt

output tooHigh :=
  alt.offset(by: -1, dft: 0) > 500
  ∧ alt > 500
  ∧ alt.offset(by: +1, dft: 0) > 500

trigger tooHigh
```

fails @ t=1

never fails

fails @ t=|σ|

Three Phases



Three Code Fragments

```
fn prefix() {
 a-1 > 500
  ∧ a0 > 500
  ∧ a+1 > 500
}
```

Prefix

```
while let Some(…)
            = get_input() {
   a-1 > 500
     ∧ a0 > 500
     ∧ a+1 > 500
}
```

Monitor Loop

```
fn postfix() {
 a-1 > 500
  ∧ a0 > 500
  ∧ a+1 > 500
}
```

Postfix

Three Code Fragments

```
fn prefix() {
   0 > 500
   ∧ a₀ > 500
   ∧ a₊₁ > 500
}
```

Prefix

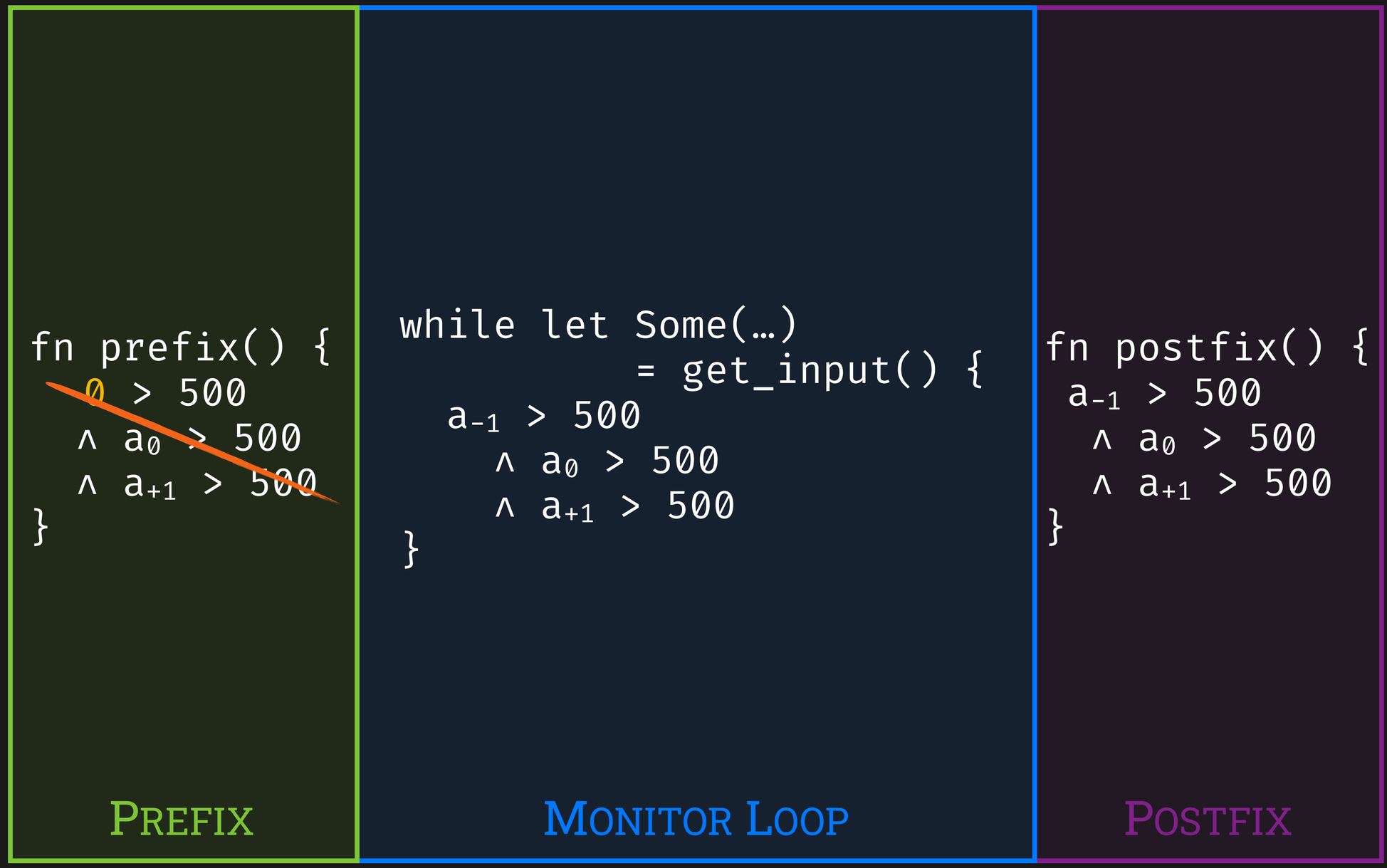
```
while let Some(…)
             = get_input() {
   a₋₁ > 500
      ∧ a₀ > 500
      ∧ a₊₁ > 500
}
```

Monitor Loop

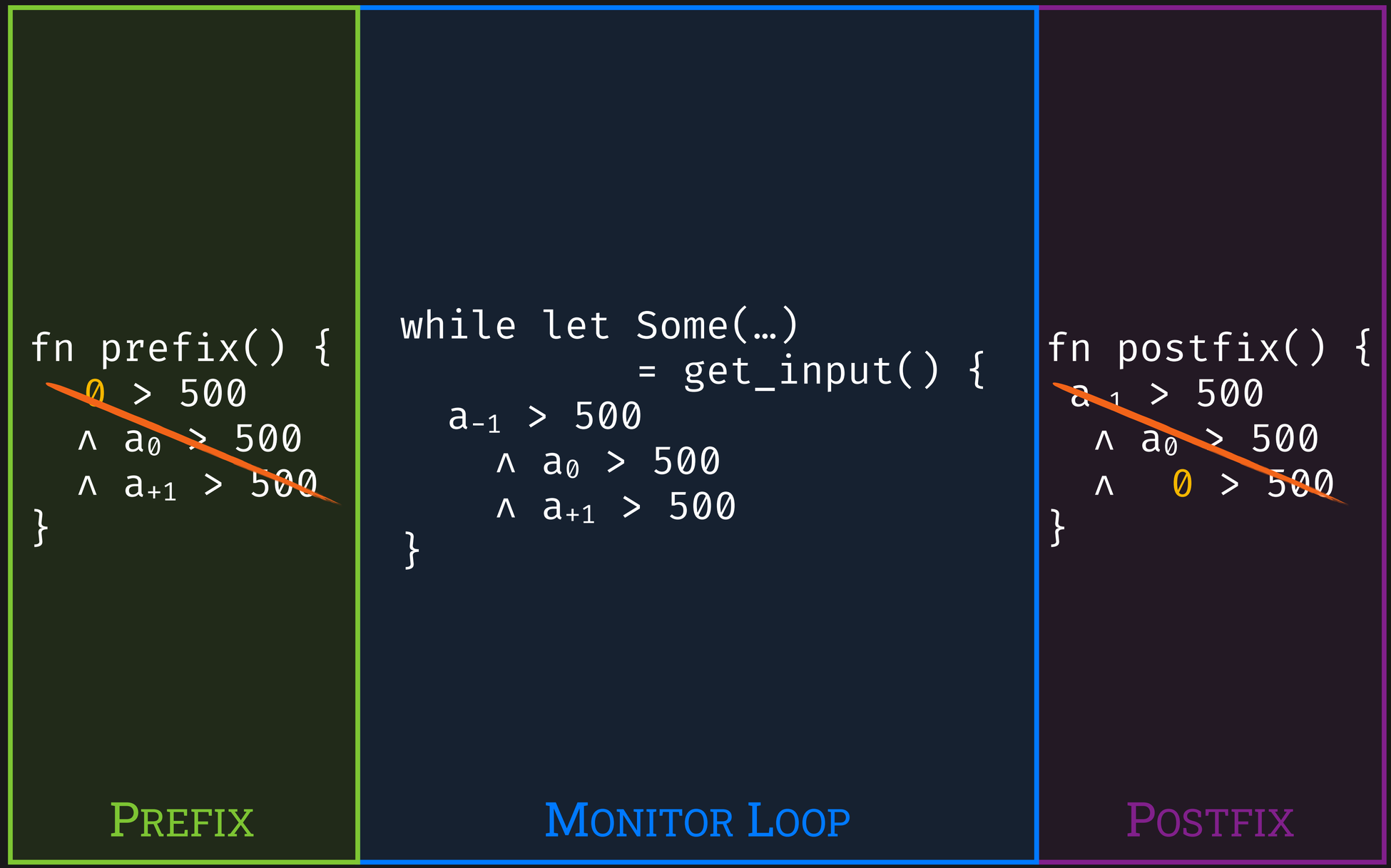
```
fn postfix() {
 a₋₁ > 500
   ∧ a₀ > 500
   ∧ a₊₁ > 500
}
```

Postfix

Three Code Fragments



Three Code Fragments



Three Code Fragments

I. Eradicate Most Conditionals
II. Memory Accesses Become Constants

```
fn prefix() {
  0 > 500
    ∧ a0 > 500
    ∧ a+1 > 500
}
```

Prefix

```
while let Some(…)
            = get_input() {
    a-1 > 500
      ∧ a0 > 500
      ∧ a+1 > 500
}
```

Monitor Loop

```
fn postfix() {
  a-1 > 500
    ∧ a0 > 500
    ∧  0 > 500
}
```

Postfix

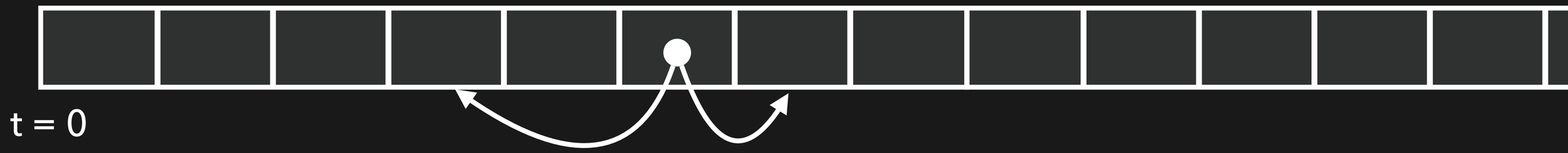
Performance Benefits

Interpretation [CAV'19]	438ns	1.535µs
Compilation	6ns (1.4%)	63ns (4%)

Verification

stream ≜ infinite sequence of values

Lola



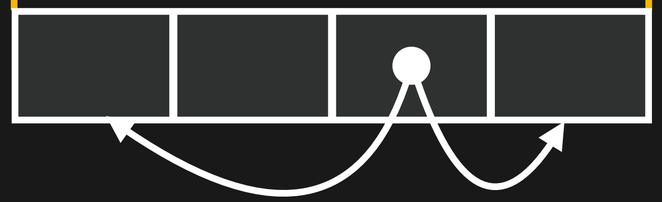
Rust

Verification

stream ≙ infinite sequence of values

Lola

Rust



memory ≙ finite excerpt of stream

Verification

stream ≜ infinite sequence of values

Lola

t = 0

Viper

t = 0

Ghost Memory

Rust

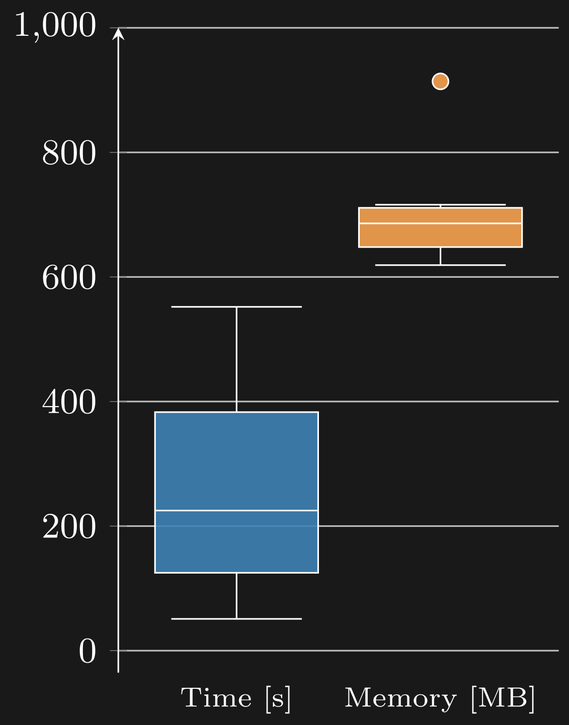
memory ≜ finite excerpt of stream

Ghost Memory in Action

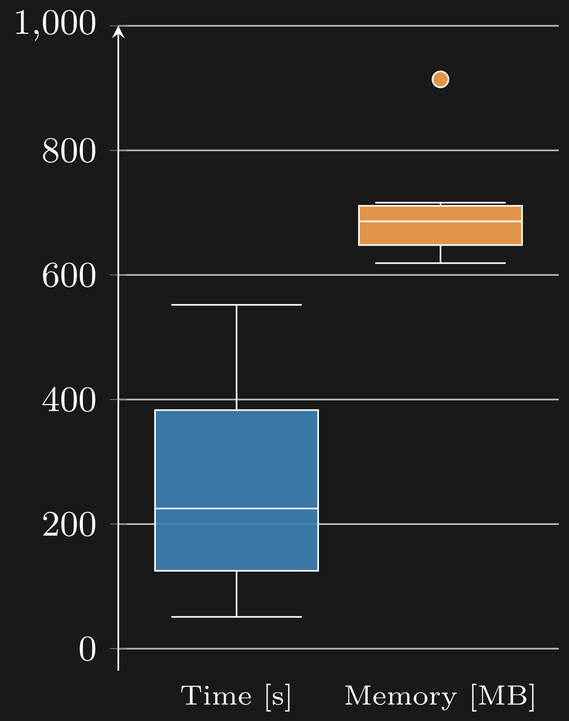
```
#[invariant="forall i: usize ::
    (0 ⩽ i && i < μ(a))
        ⟹ mem.get_a(i) == gm.get_a(iter - i)
"]

#[invariant="new_tooHigh == gm.get_a(iter - 2) > 500 ∧ …"]

while let Some(input) = get_input() {

    mem.add_input(&input);

    [[ EVALUATION LOGIC ]]

    mem.store(new_tooHigh);
    gm.store(new_tooHigh);

    if trigger_1 { emit( trigger_1_msg) }
}
```

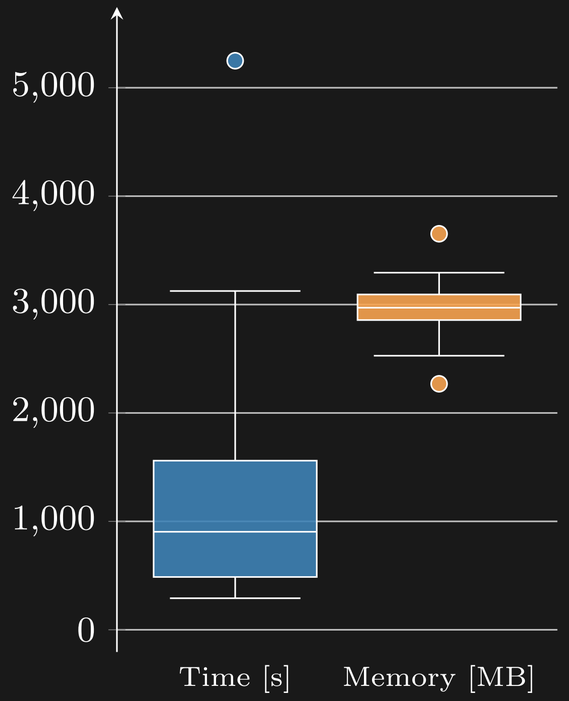
Experiments

Experiments



Experiments

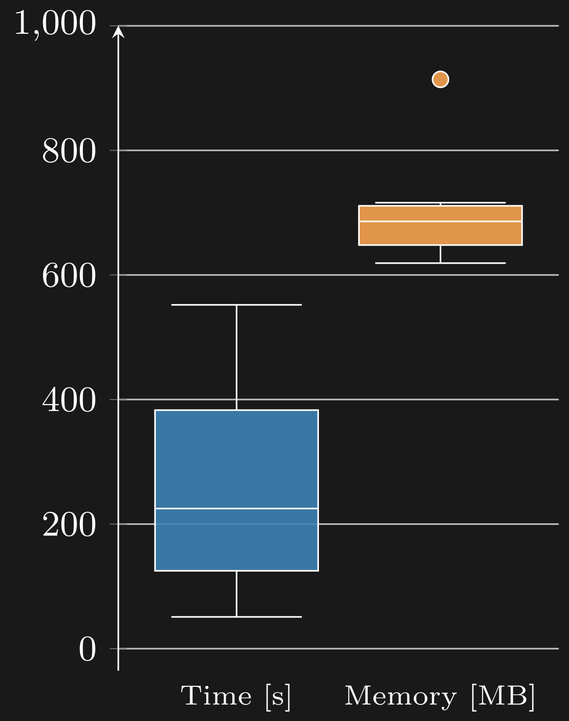


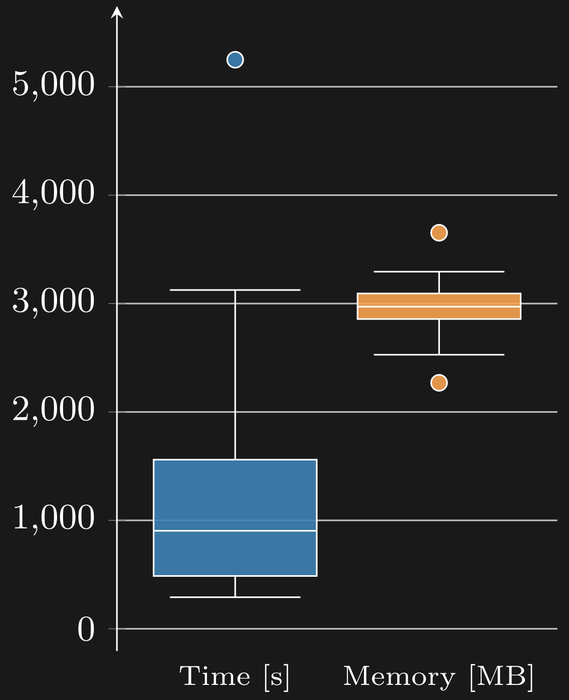


Experiments

altitude

flight phase

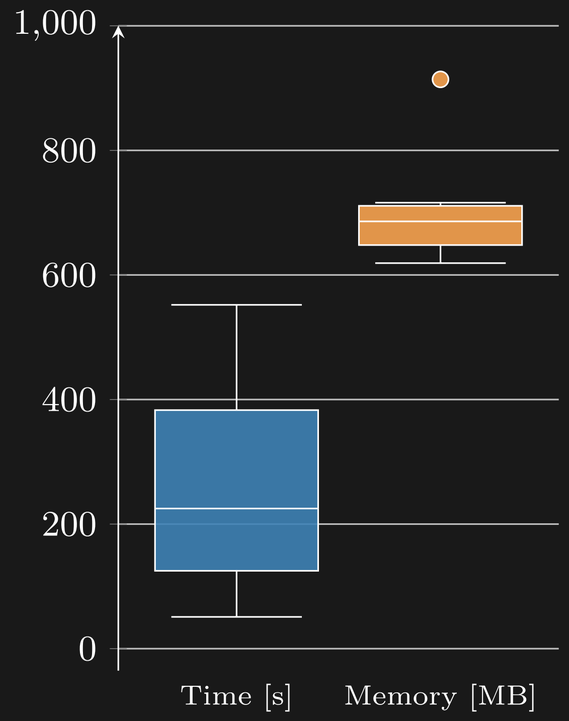


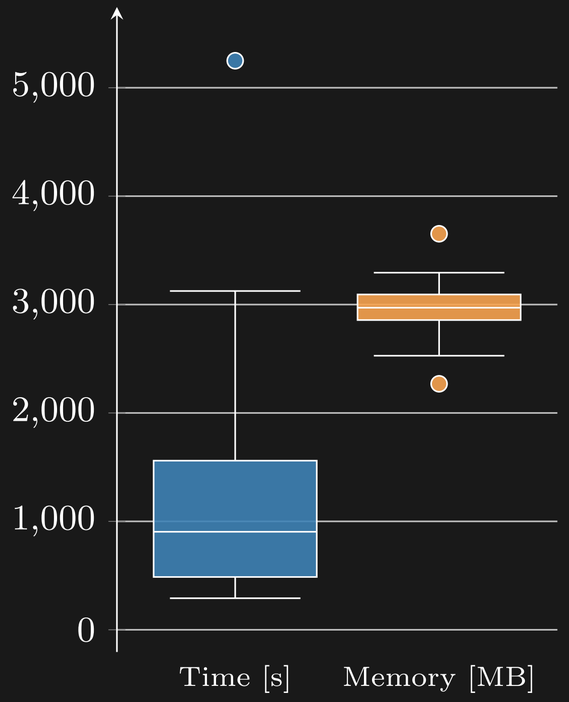


Experiments

altitude

flight phase



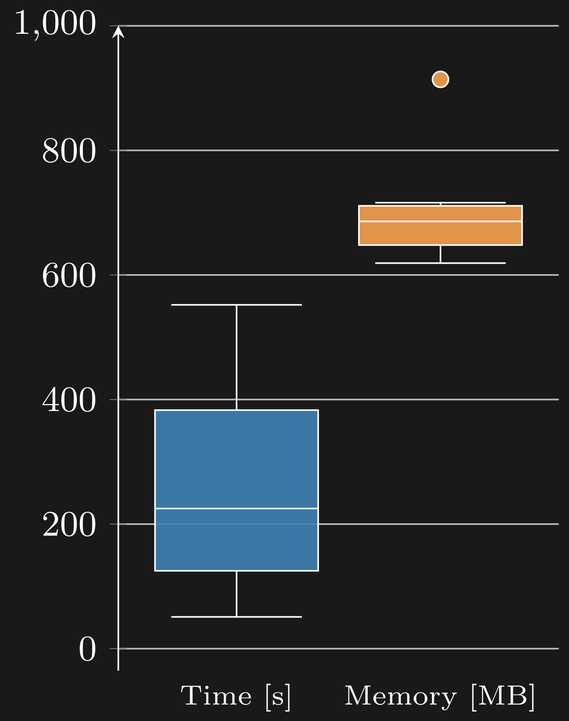


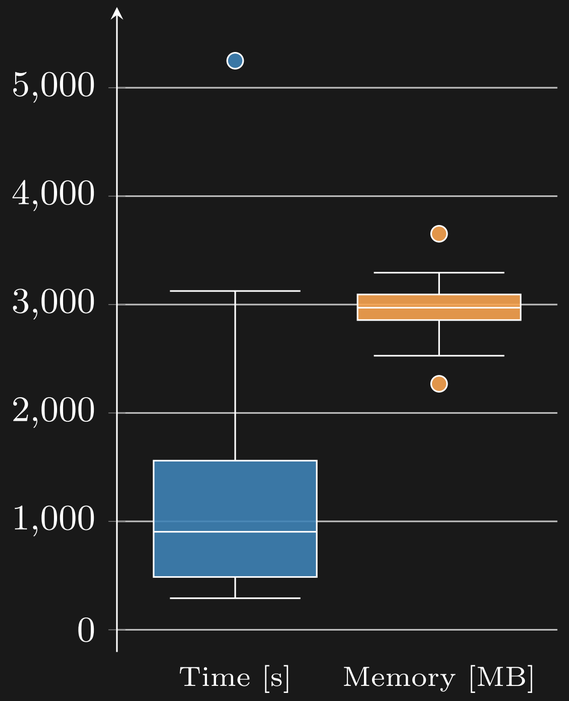
Detected implicit assumption on input stream!

Experiments

altitude

flight phase





Detected implicit assumption on input stream!

On corrected spec:

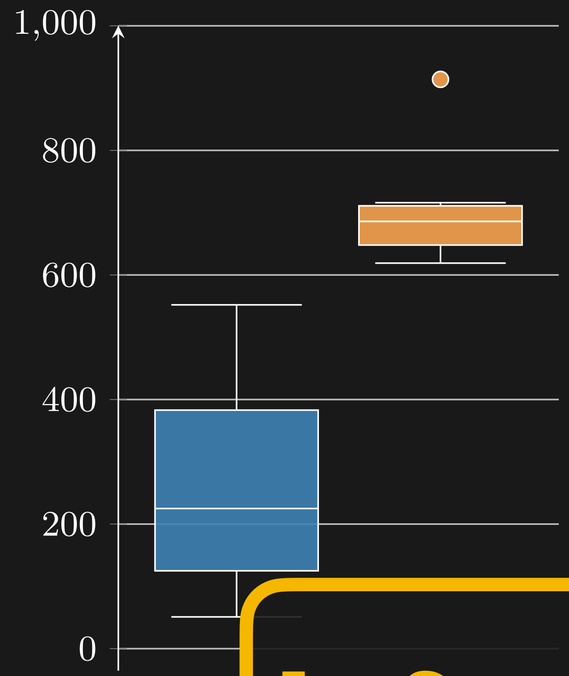
6 – 16min
1.38 – 1.66GB

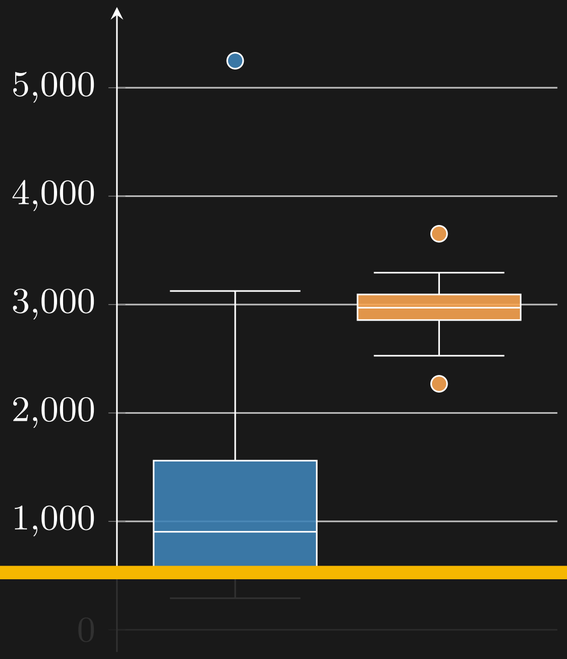
2 T/O (10%)
4 fails (20%)

Experiments

altitude

flight phase





Detected implicit assumption on input stream!

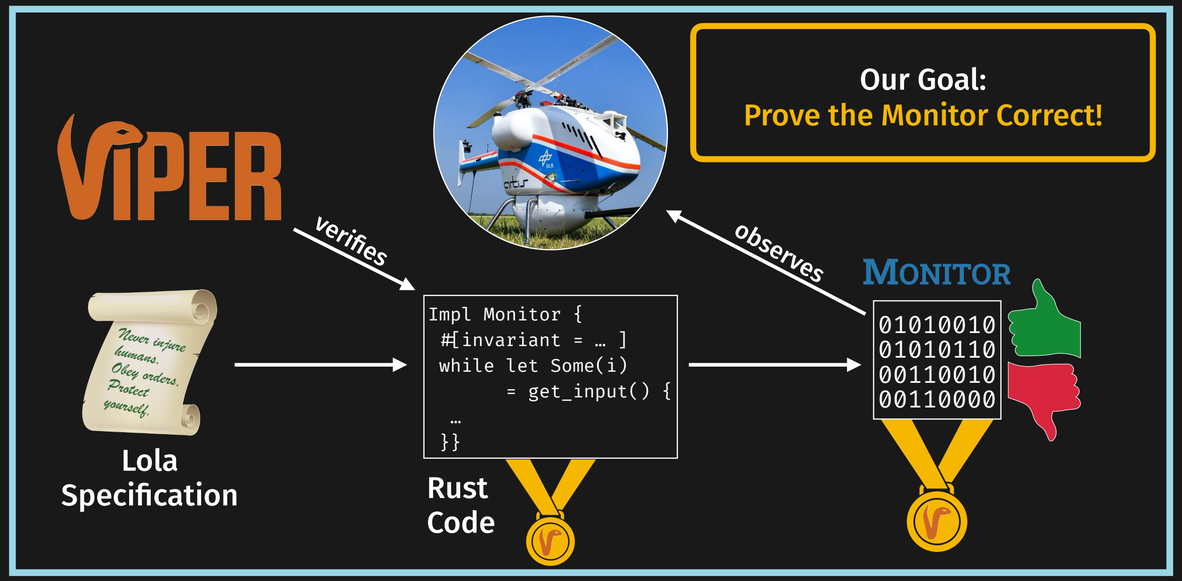
On corrected spec:

6 – 16min

1.38 – 1.66GB

I. Successfully detected specification error
II. Verified monitors for complex specifications

Conclusion



More information:

Compiled Code:

- Github → Lola2RustArtifact

In Paper:

- Specification Analysis
- Concurrent Implementation

Bottom Line:

- Quis Custodiet Ipsos Custodes? Viper does!
- Successful verification
- Highly performant Code