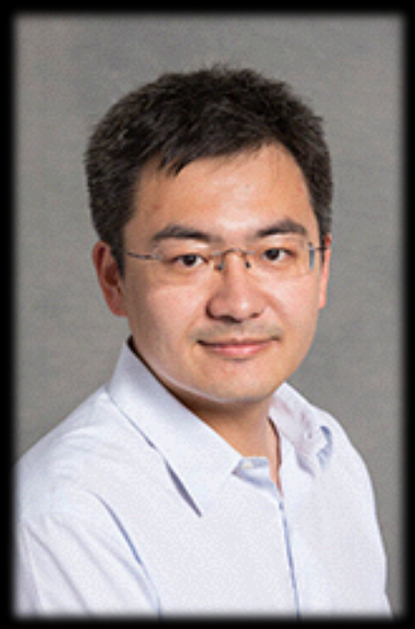
CREST

# **GI** using Deep Parameter Tuning

Fan Wu

Wes Weimer

Mark Harman



Yue Jia

Jens Krinke

# Why GI for non-functional properties

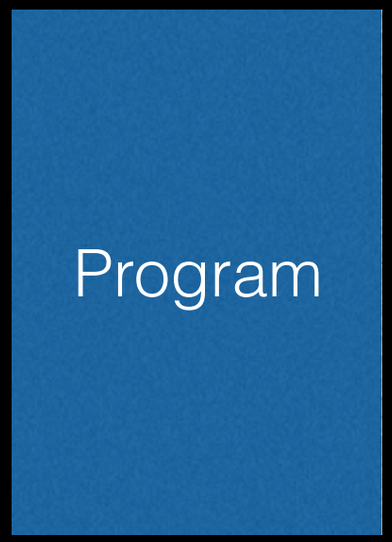
Functional Requirements

humans have to define these

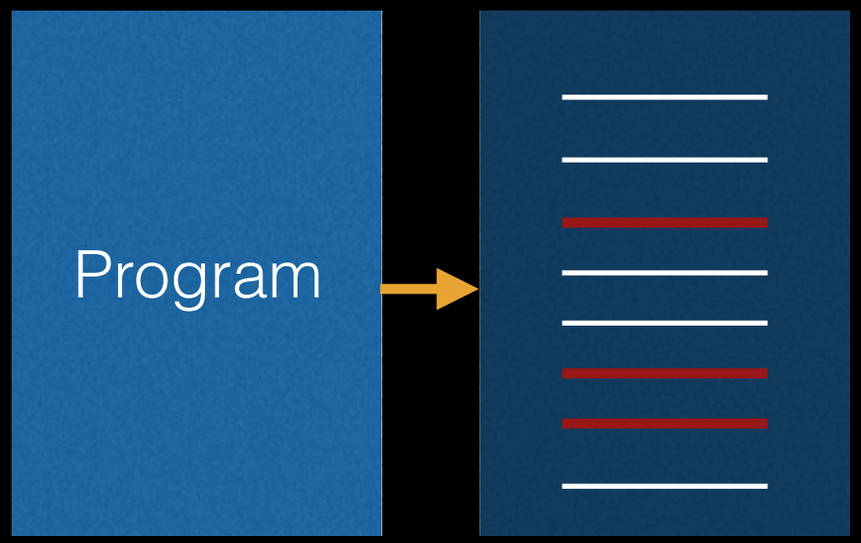
Non-Functional Requirements

a machine can optimise these

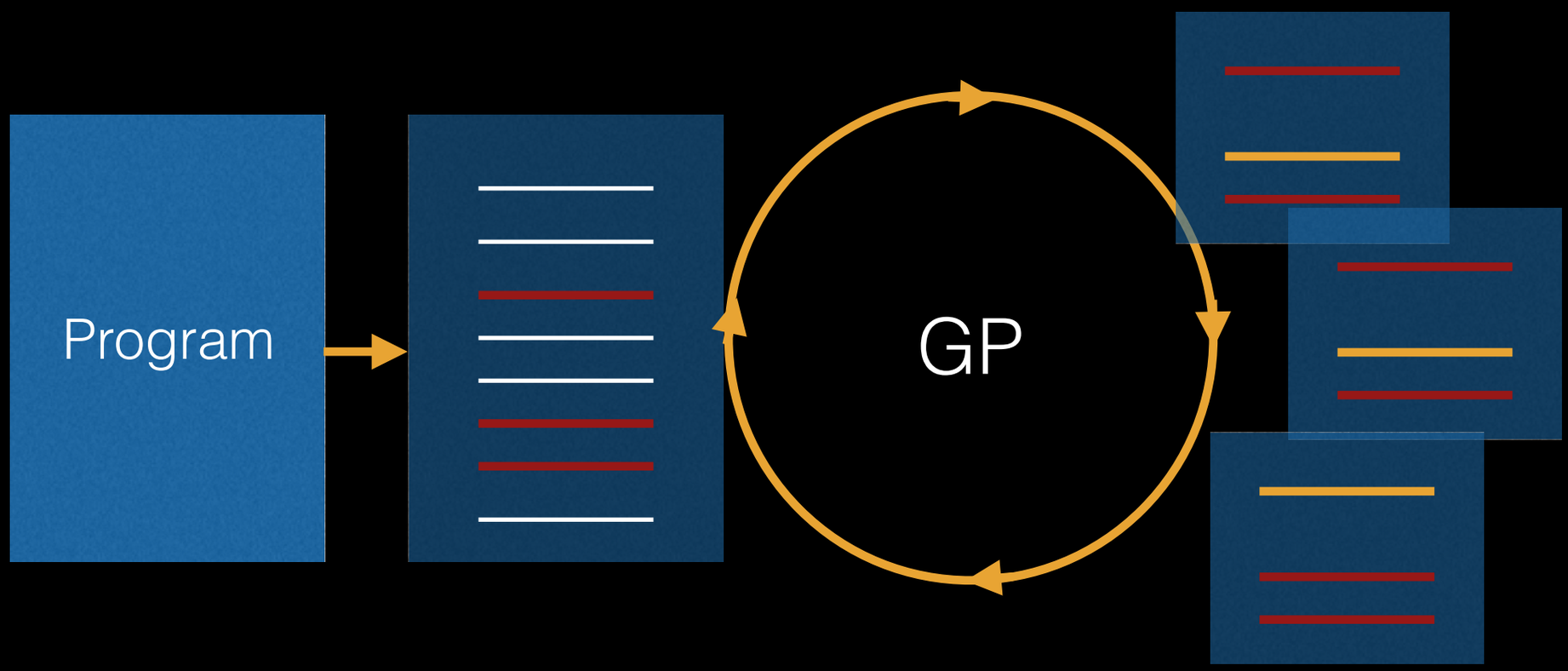
# Traditional GI Process



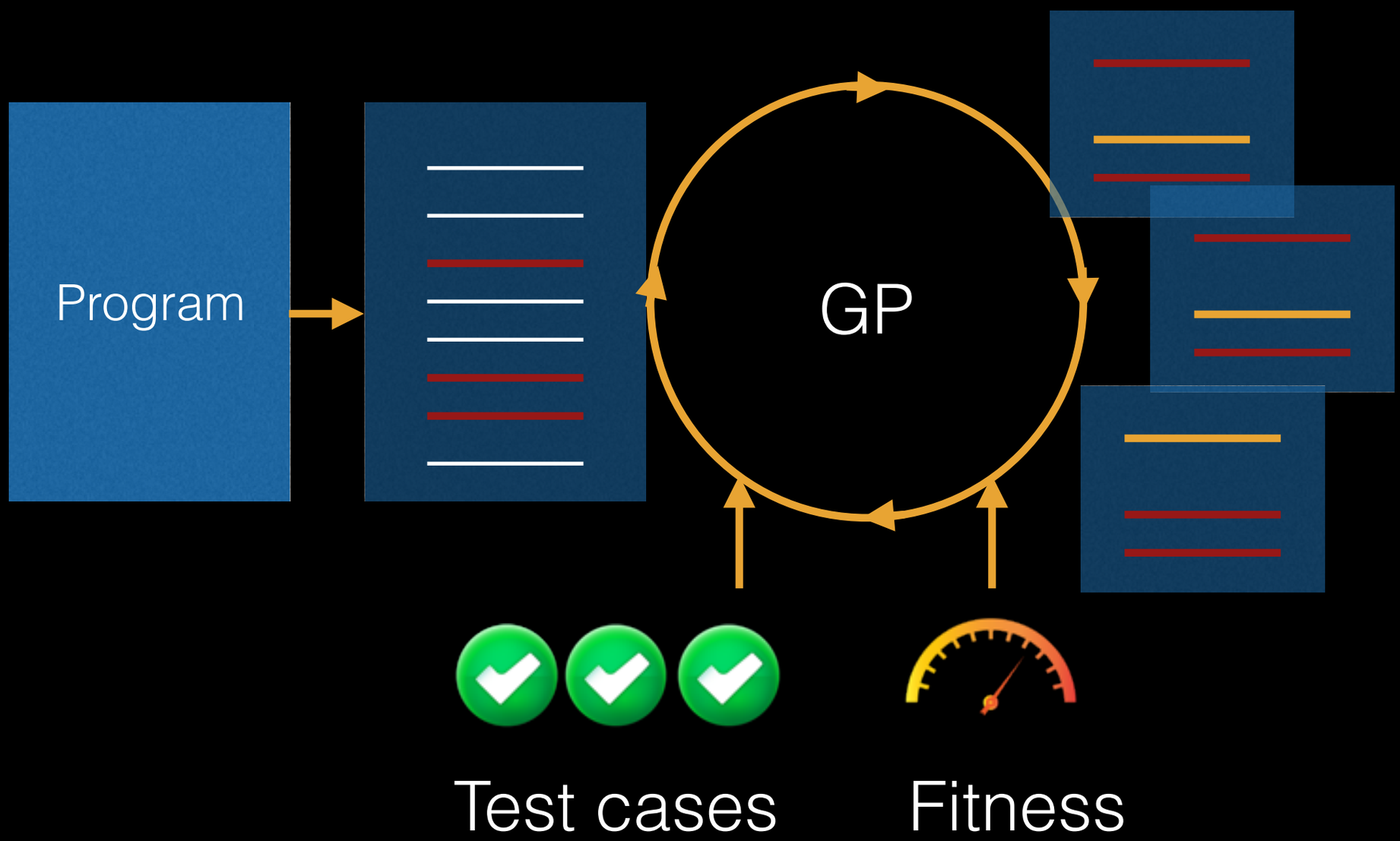
# Traditional GI Process



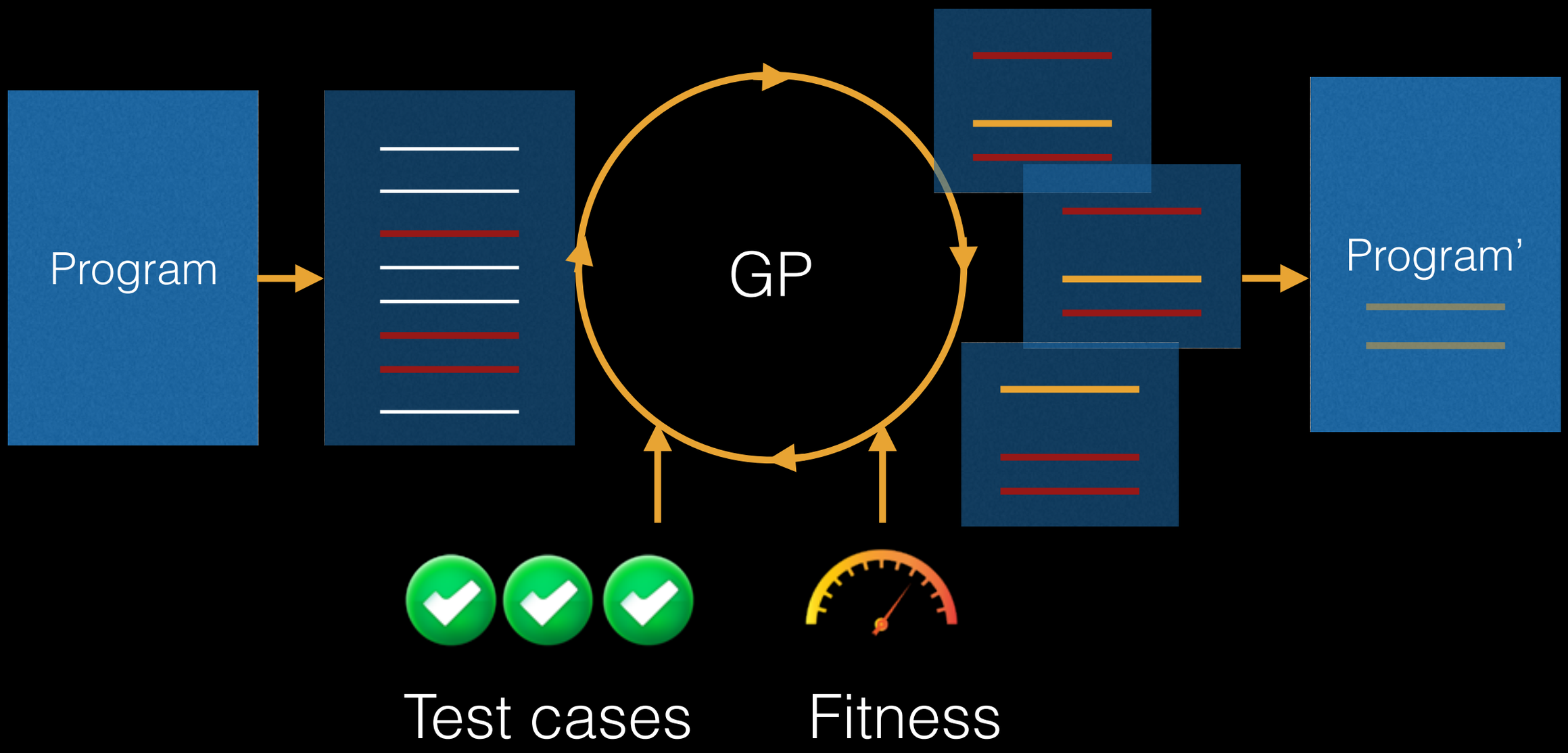
# Traditional GI Process



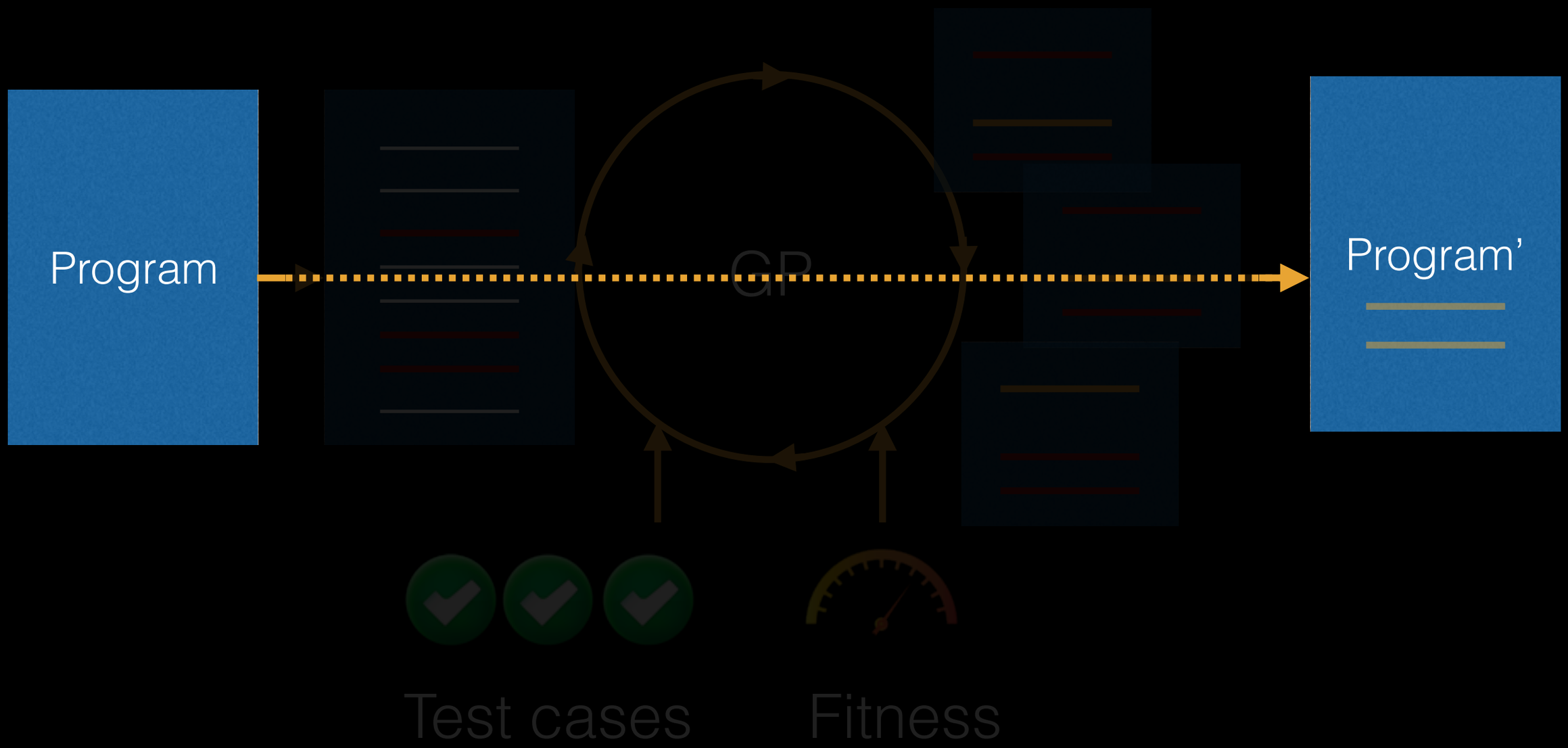
# Traditional GI Process



# Traditional GI Process

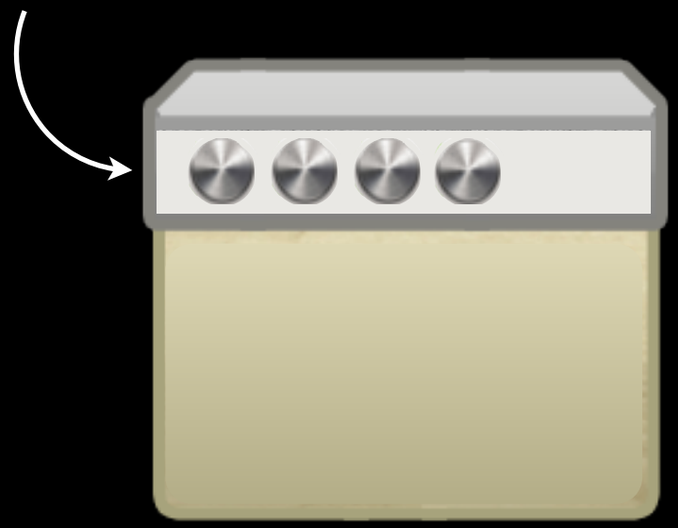


# Traditional GI Process

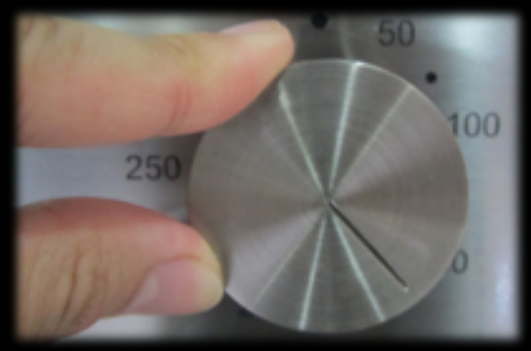


# Parameter Tuning

Tuneable Parameters



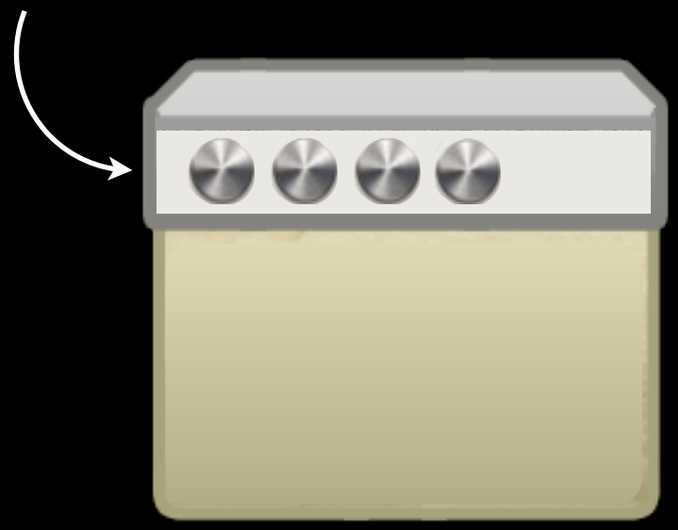
Program



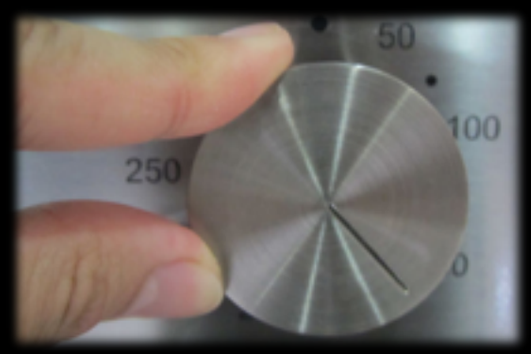

e.g. GCC -O3 foo.c

# Parameter Tuning

Tuneable Parameters



Program




e.g. GCC -O3 foo.c

**Good alternative for GI?**

# Parameter Tuning

Tuneable Parameters

Program

Not Many Tuneable Parameters

Limited power

**Good alternative for GI?**

# Parameter Tuning

Tuneable Parameters

Program

Many software systems contain undocumented internal variables or expressions that also affect the performance of the systems.

# Parameter Tuning

Tuneable Parameters

Program

Our approach

Tuneable Parameters

Program

Tuneable Implicit Parameters

Many software systems contain undocumented internal variables or expressions that also affect the performance of the systems.

# Deep Parameter Tuning

Tuneable Parameters **(Shallow Parameters)**

Program

Tuneable Implicit Parameters

**(Deep Parameters)**

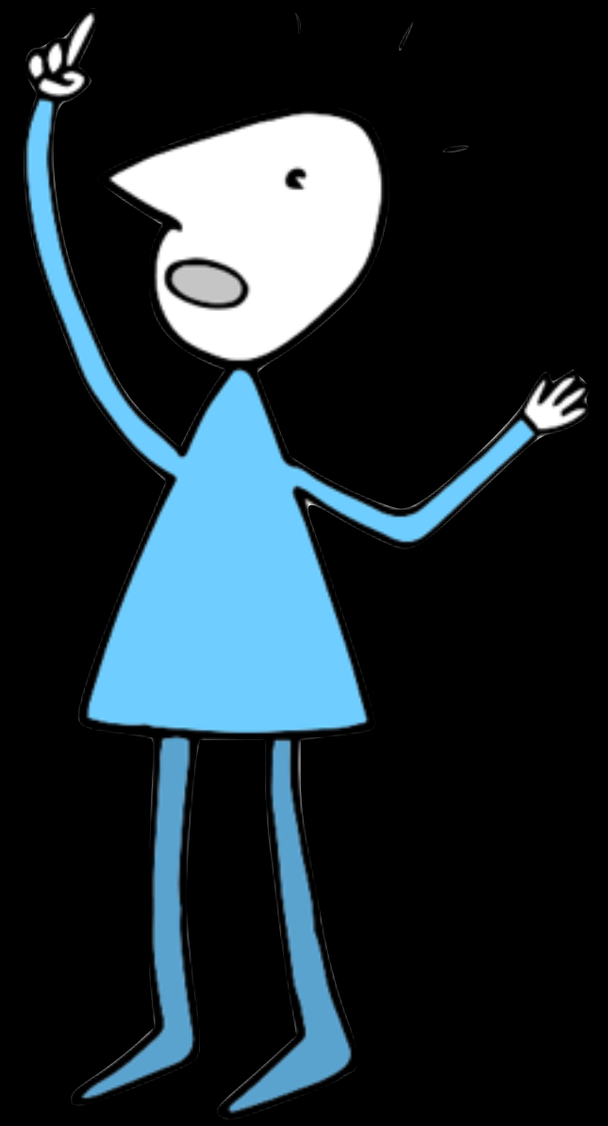
**NSGAII**

**Opentuner**

**Dynamic Knobs**

# Is this GI?

Genetic Improvement (GI) aims to find improved versions of existing programs that retain some partial semantics of the original (and possibly some of its syntax too)



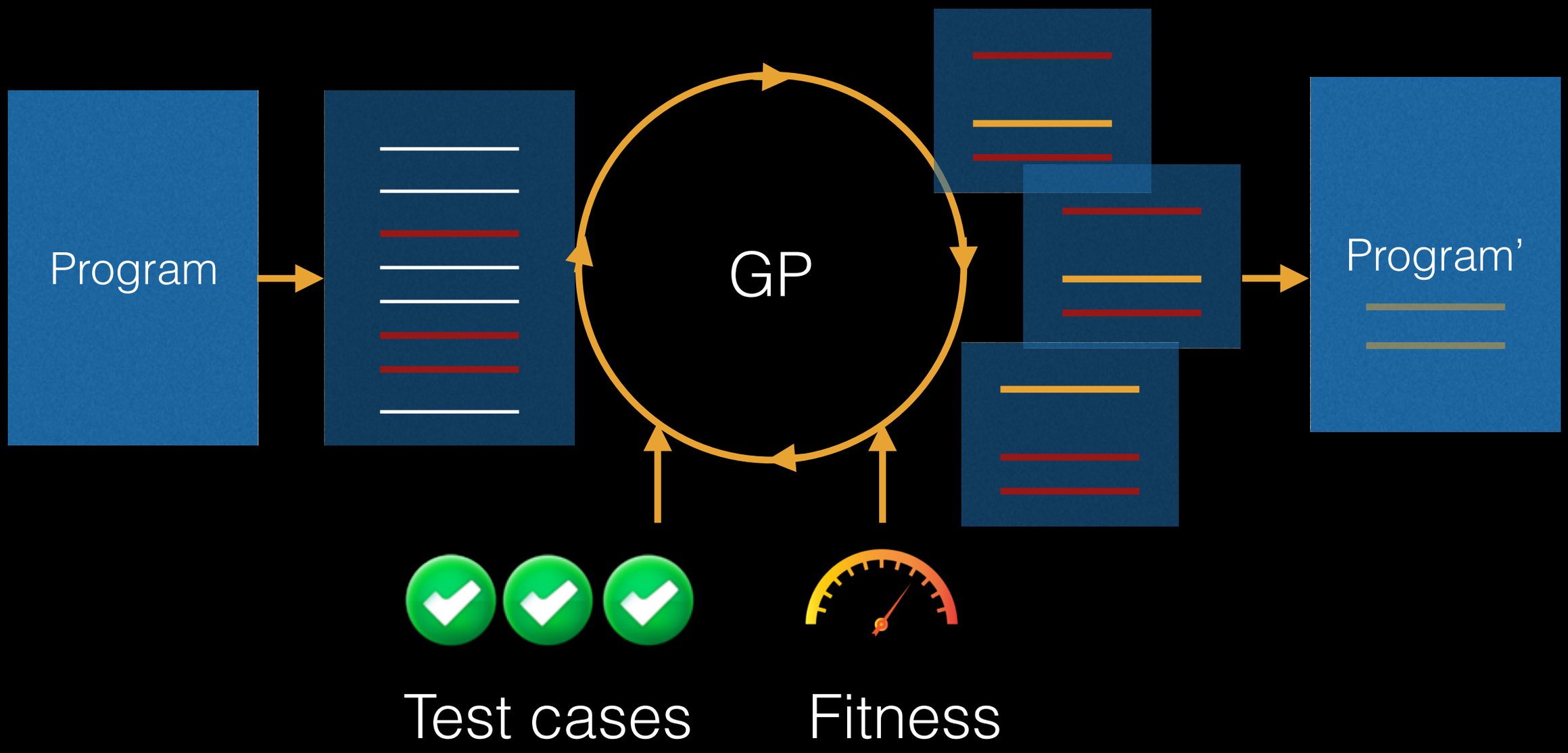
# Comparing to GI


Easy to use

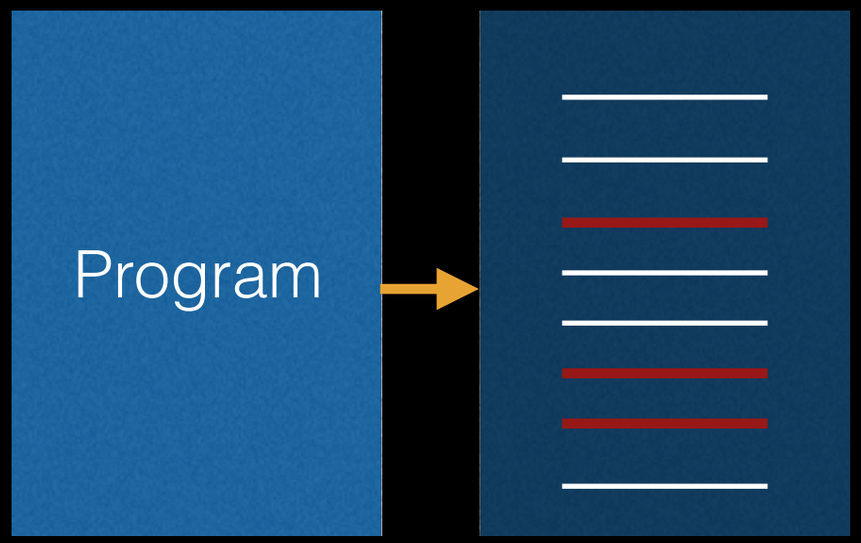
Lightweight

Less fragile

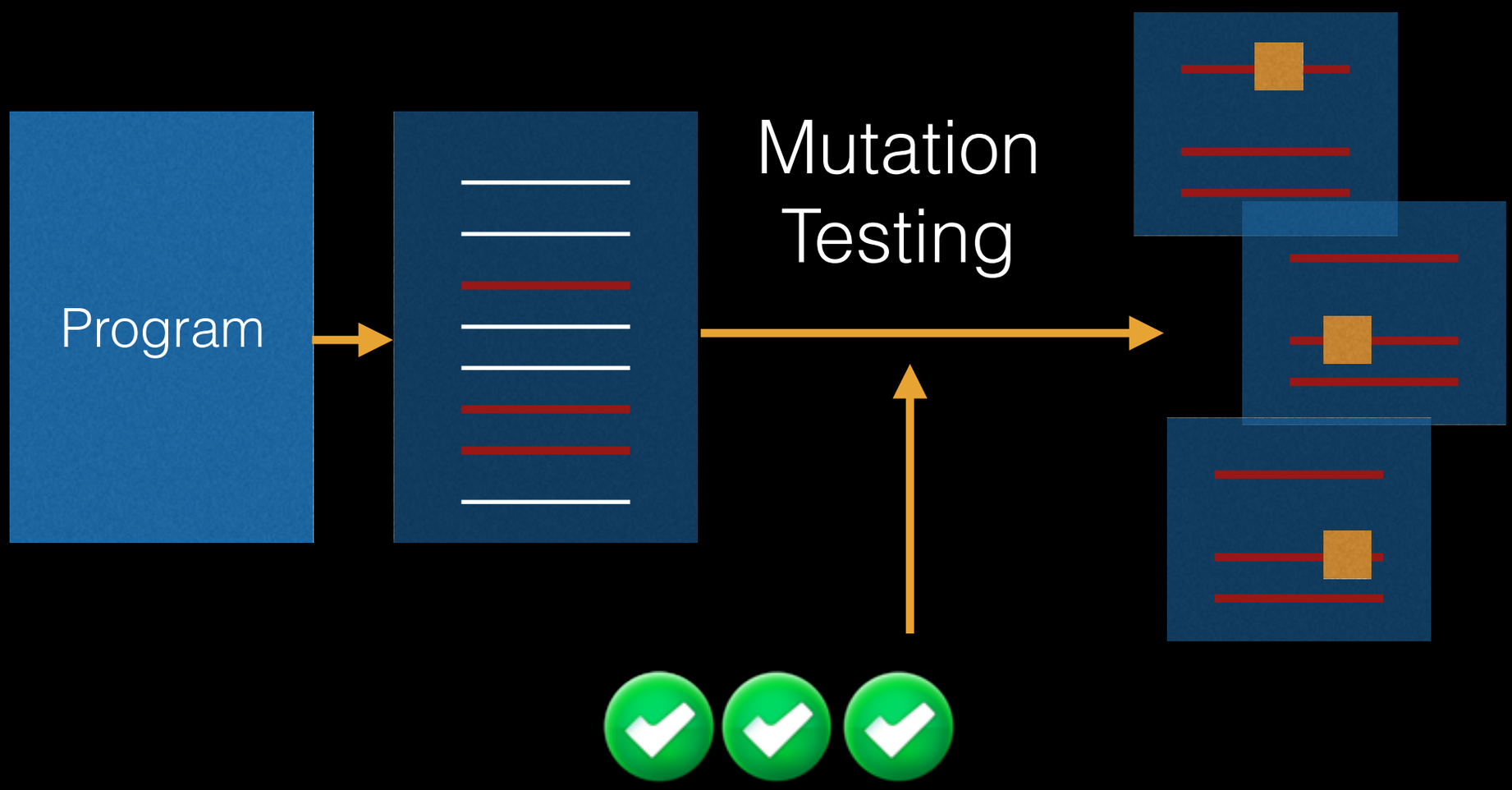
# Our approach



# Our approach



# Our approach



# Mutation Operators

## C Mutation Testing tool : Milu

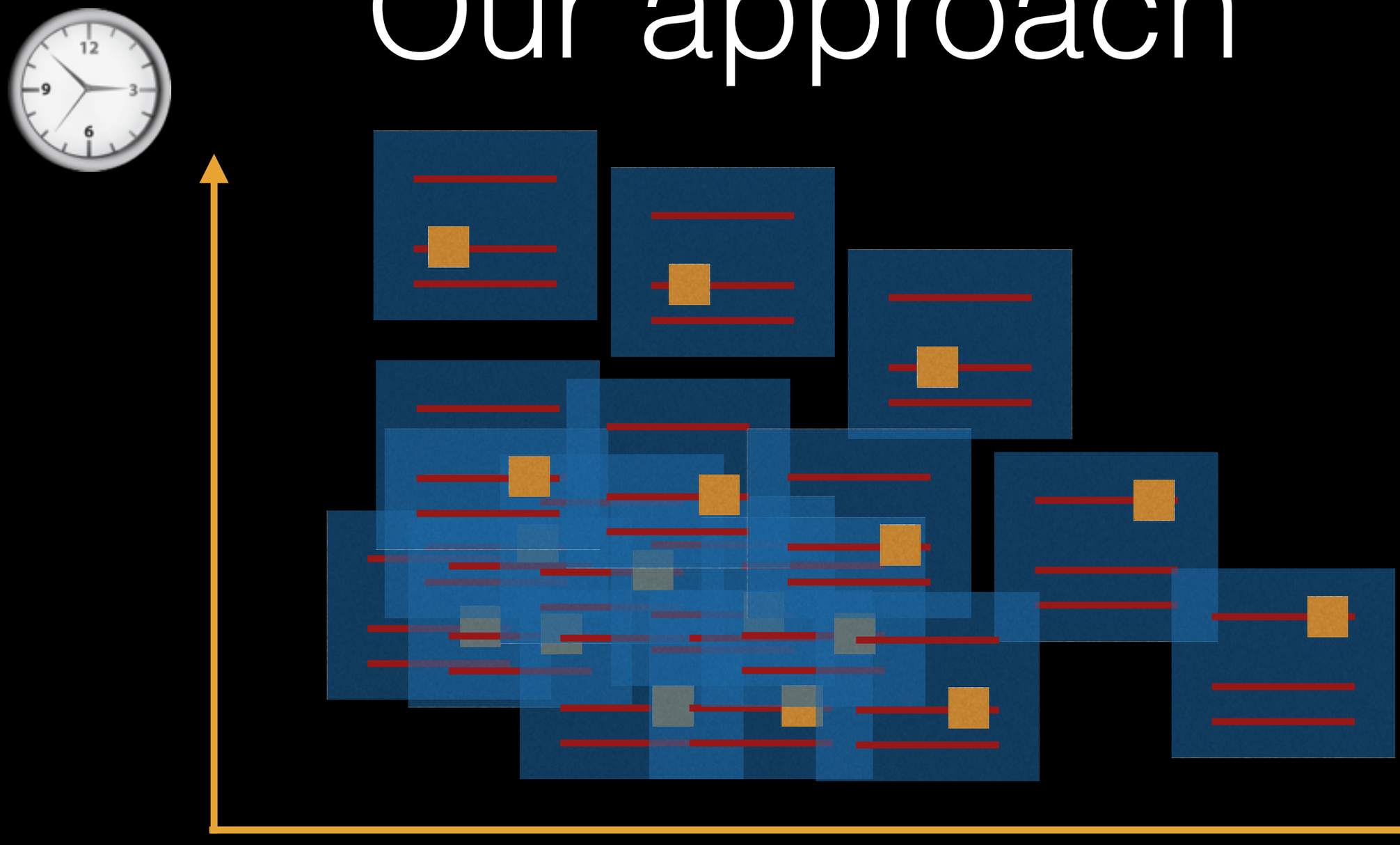
**Table 1: Selected mutation operators**

| Mutation Operators | Changes Between |
|---|---|
| CRCR – Constant replacement | constants, 0, 1, −1 |
| OAAN – Arithmetic operator | `+`, `-`, `*`, `/`, `%` |
| OAAA – Arithmetic assignment | `+=`, `-=`, `*=`, `/=`, `%=` |
| OCNG – Logical context negation | *expr*, `!`*expr* |
| OIDO – Increment/decrement | `++x`, `--x`, `x++`, `x--` |
| OLLN – Logical operator | `&&`, `\|\|` |
| OLNG – Logical negation | $x$ `op` $y$, $x$ `op` `!`$y$, `!`$x$ `op` $y$, `!(`$x$ `op` $y$`)` |
| ORRN – Relational operator | `>`, `>=`, `<`, `<=`, `==` |
| OBBA – Bitwise assignment | `&=`, `\|=` |
| OBBN – Bitwise operator | `&`, `\|` |

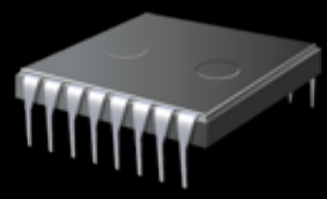
# Our approach


**Pseudo Equivalent Mutants:**
**Mutants are not killed by any tests**

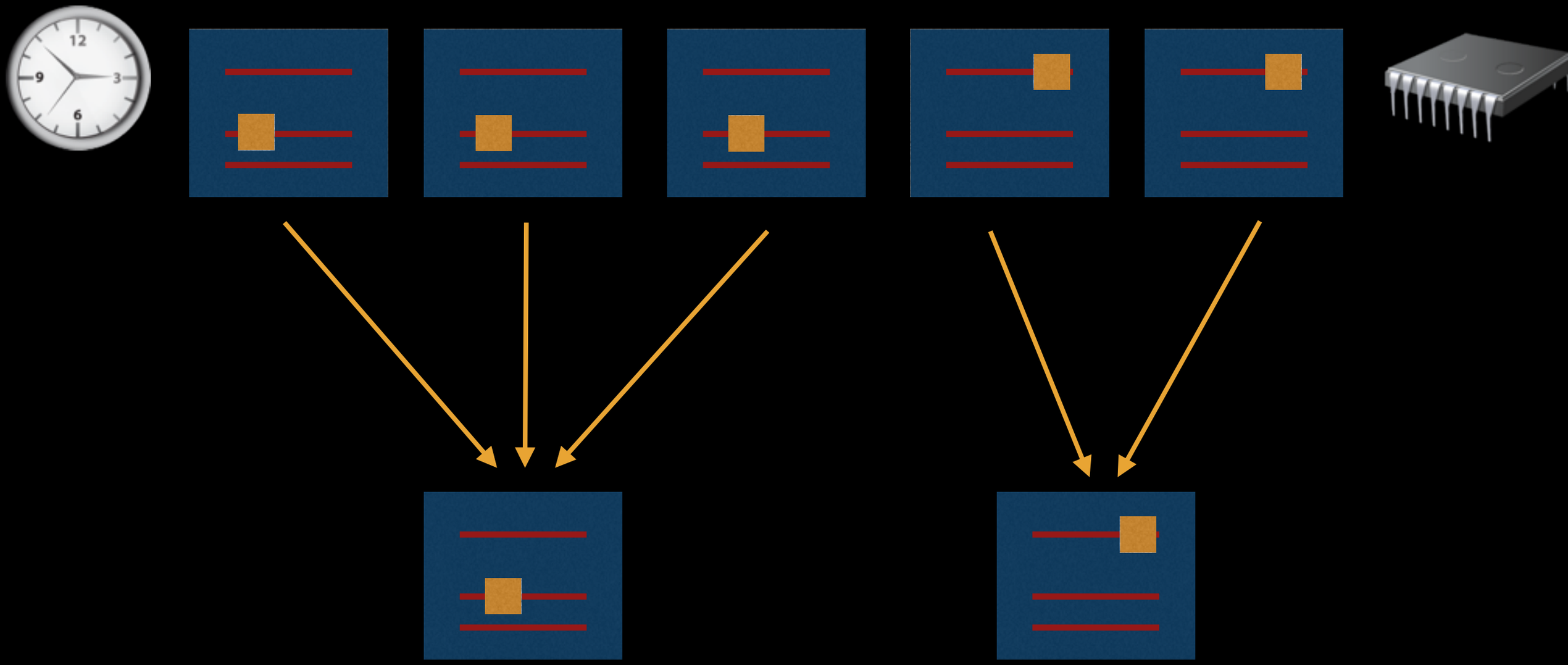
# Our approach



**Non-dominated Rank**
**Pseudo Equivalent Mutants**



# Our approach



**Exposing Deep Parameters**

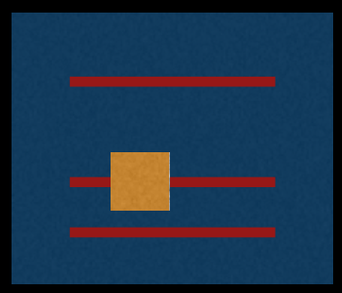
# Our approach

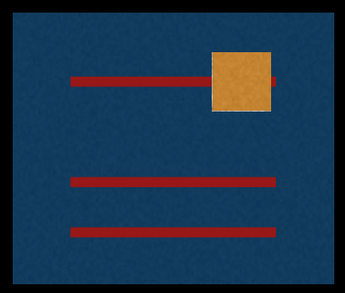
Given a location L, and $E_L$ is the expression at L

$$E_L \rightarrow \begin{cases} (E_L + v_L) \\ (E_L)\ xor\ v_L \end{cases}$$

arithmetic expression

logical expression





**Exposing Deep Parameters**

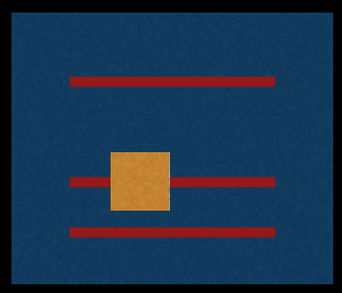
# Our approach

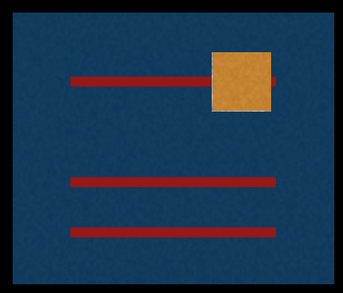
Given a location L, and $E_L$ is the expression at L

$$E_L \rightarrow \begin{cases} (E_L \; v_L) \\ (E_L) \; xor \; v_L \end{cases}$$

arithmetic expression

logical expression

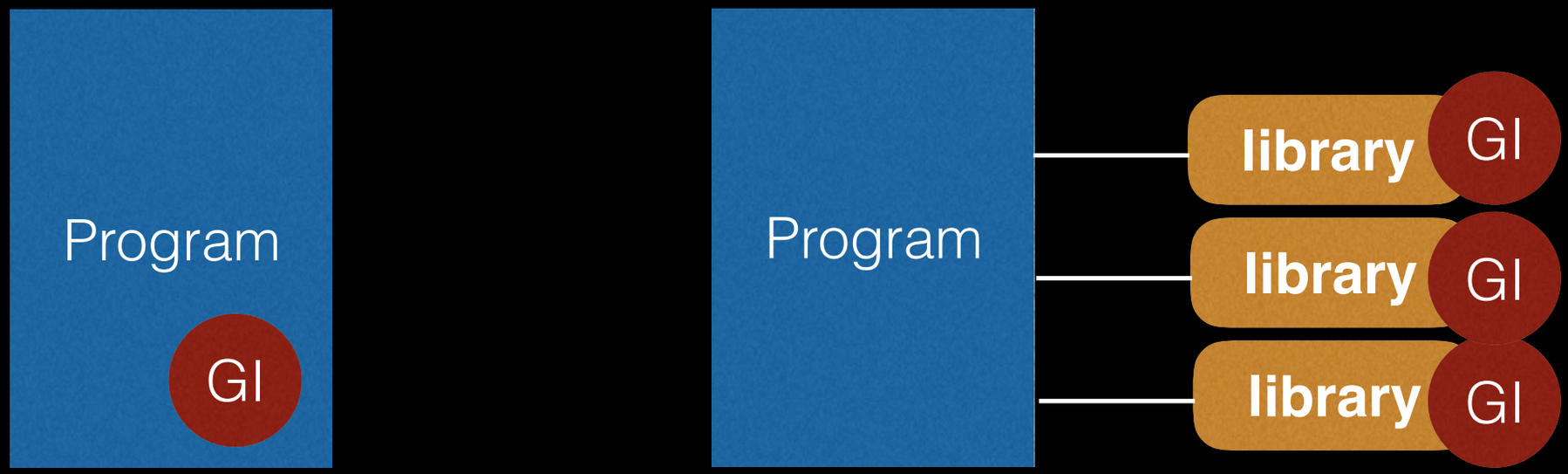




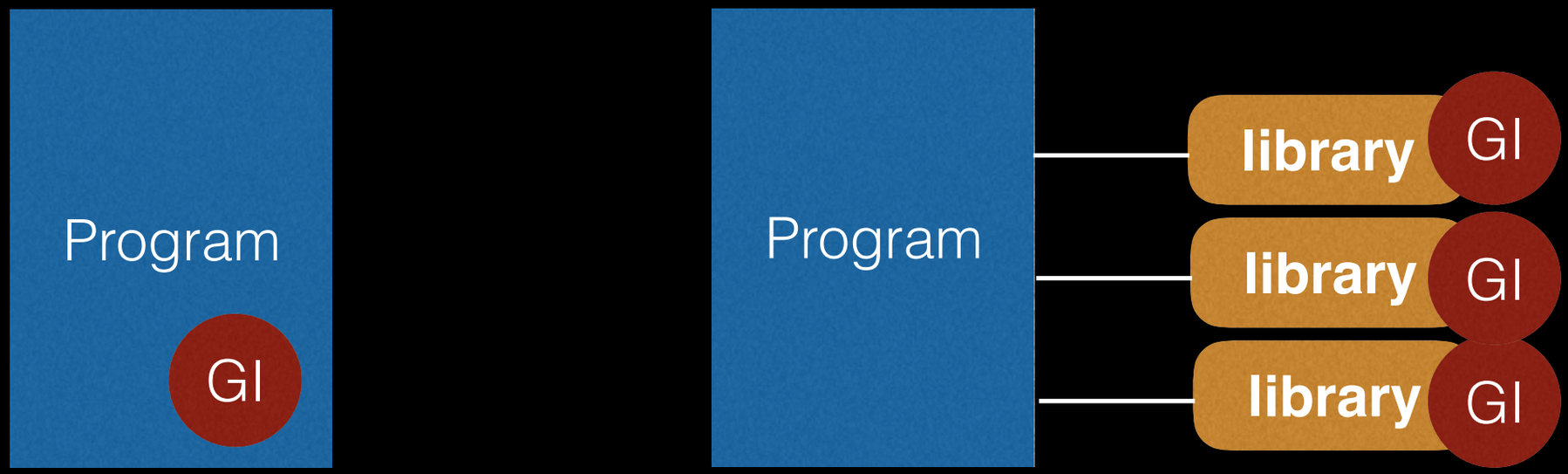
**Exposing Deep Parameters**

# Why to apply GI?

# Why to apply GI?



# Why to apply GI?




**General purpose memory allocator dlmalloc by Doug Lea**

# Experiment Subjects

**Table 3: Subject applications**

| Name | Loc | # Tests | Description |
|---|---|---|---|
| espresso | 13256 | 19 | Digital circuit simplification |
| gawk | 45241 | 334 | String processing |
| flex | 9597 | 62 | Fast lexical analyzer generator |
| sed | 5720 | 362 | Special file editor |

# Experiment Settings

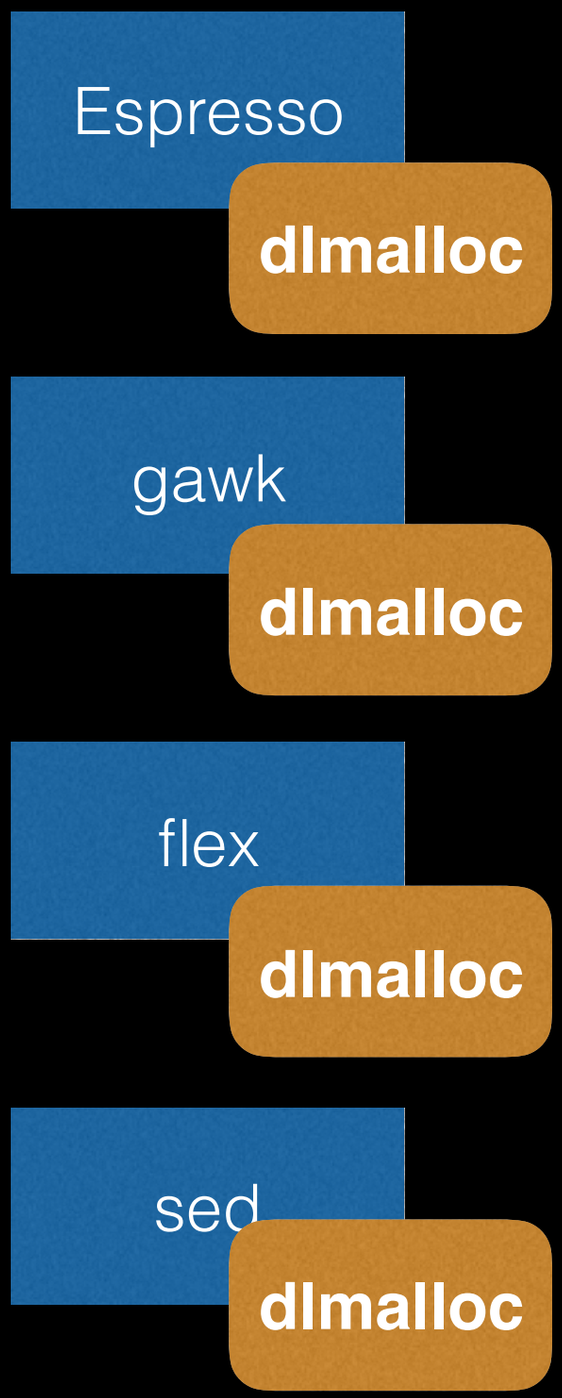
9 Deep Parameters

Repeat 10 times (cpu time and memory)

Repeated 20 times

| Name | Default |
|---|---|
| MALLOC_ALIGNMENT | $2*sizeof(void*)$ |
| FOOTERS | *false* |
| INSECURE | *false* |
| NO_SEGMENT_TRAVERSAL | *false* |
| MORECORE_CONTIGUOUS | *true* |
| DEFAULT_GRANULARITY | 0 |
| DEFAULT_TRIM_THRESHOLD | 2048 KB |
| DEFAULT_MMAP_THRESHOLD | 256 KB |
| MAX_RELEASE_CHECK_RATE | 4095 |

# Experiments




Shallow Parameters Only

---

Deep + Shallow Parameters Only

# Experiments

Espresso

**dlmalloc**

gawk

**dlmalloc**

flex

**dlmalloc**

sed

**dlmalloc**

**Random**

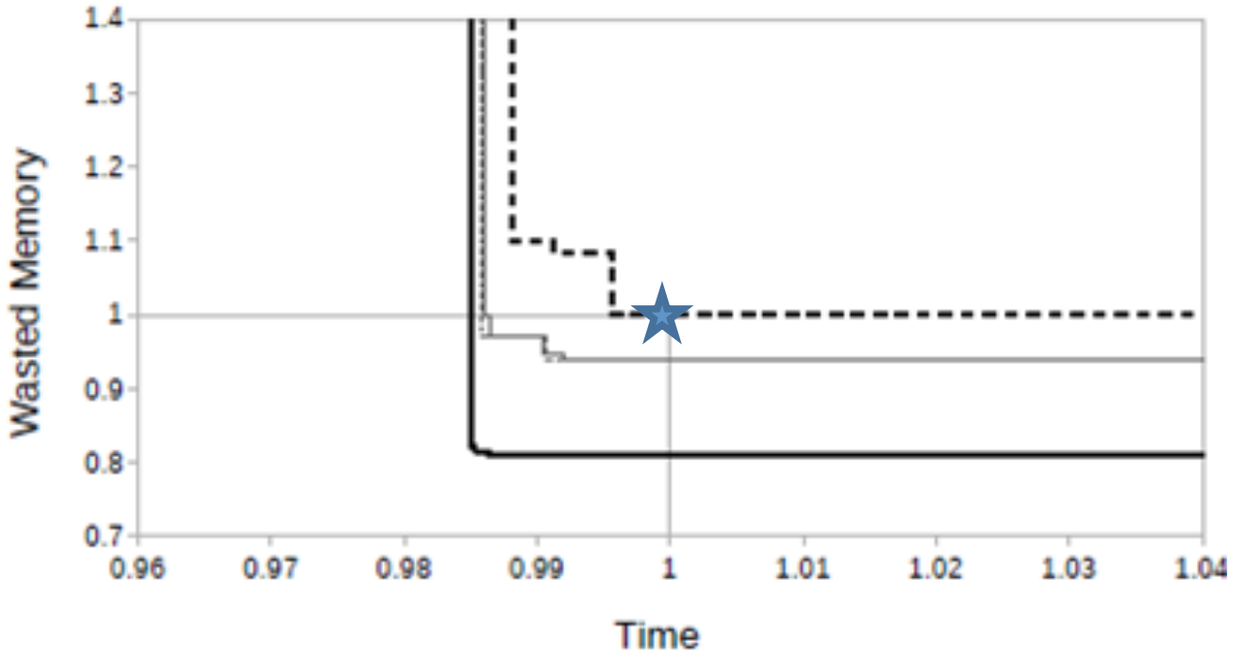
**NSGAII**

Shallow Parameters Only

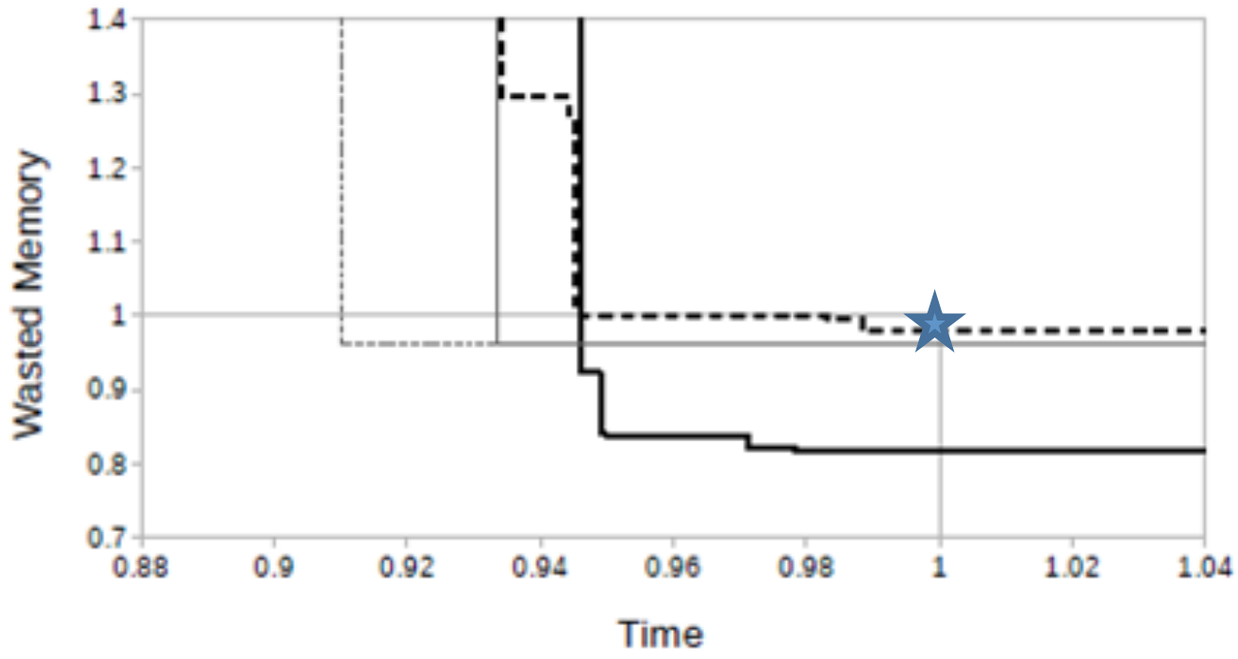
---

Deep + Shallow Parameters Only

# Results : Optimisation Quality



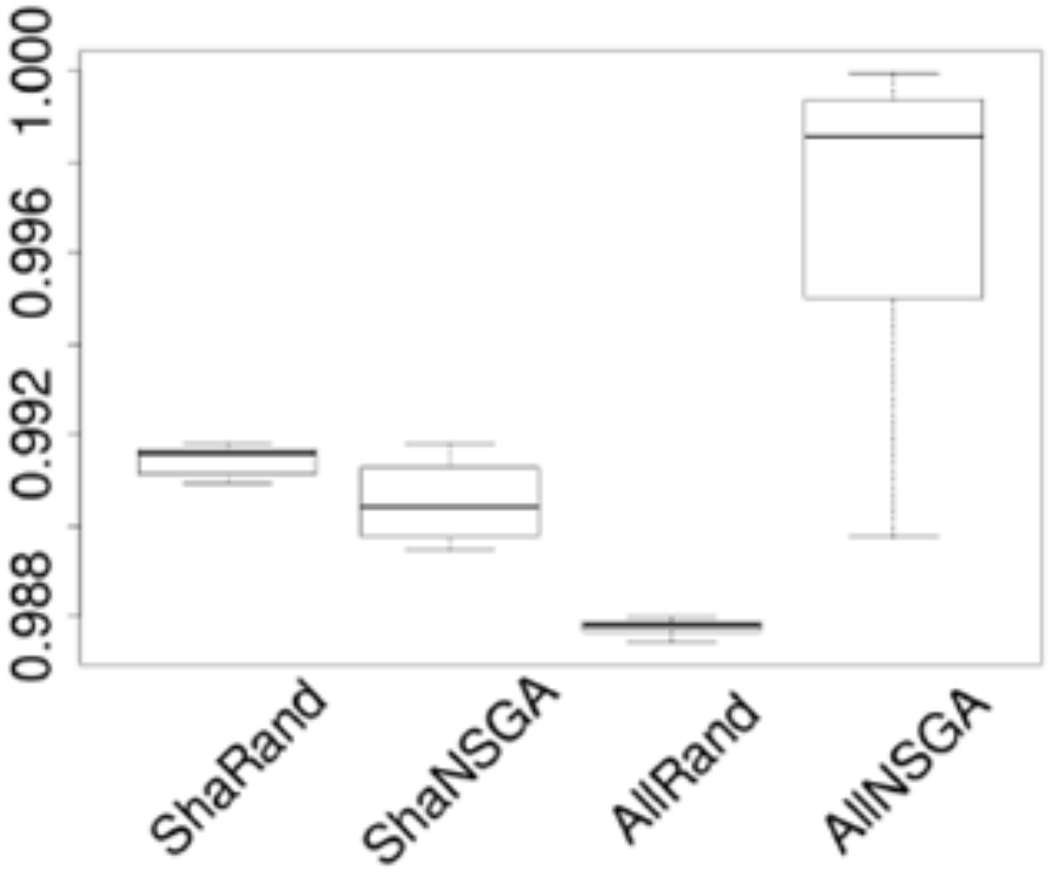

(a) espresso



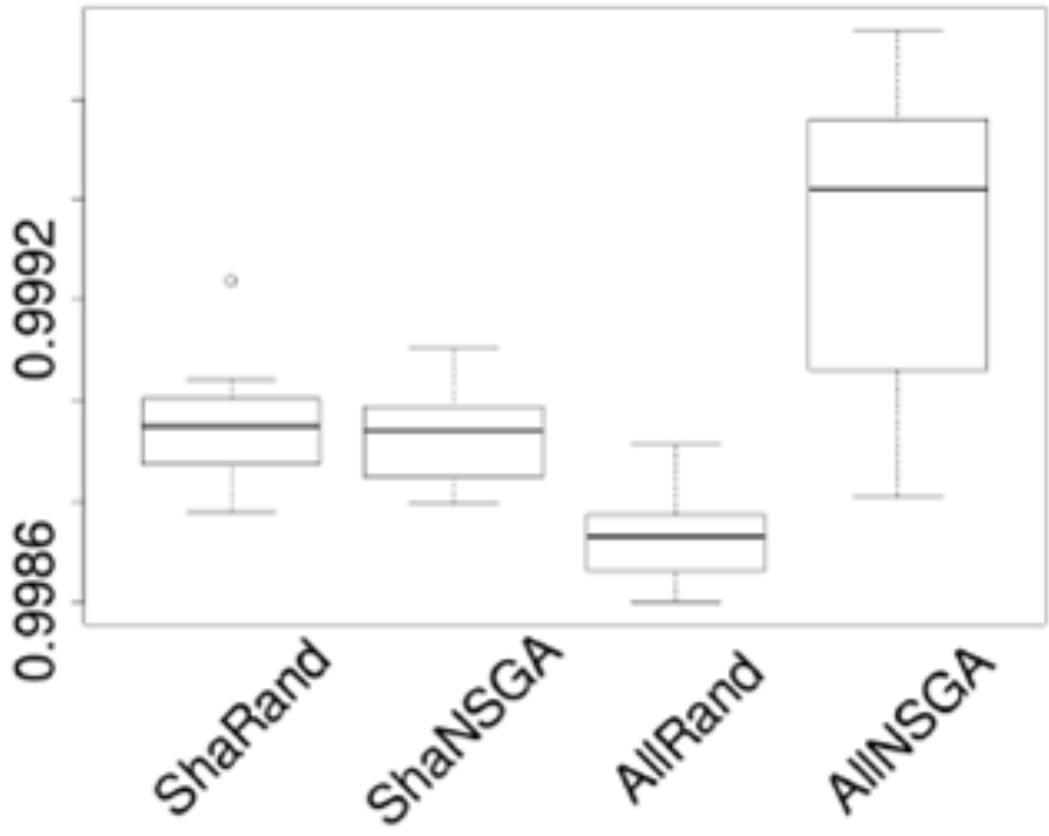

(d) sed

ShaRand ShaNSGA AllRand AllNSGA

# Results : Optimisation Quality




(a) espresso




(d) sed

# Results: Reduction

| Subject | Time Original (s) | Time Reduction (%) | | | |
|---|---|---|---|---|---|
| | | *ShaRand* | *ShaNSGA* | *AllRand* | *AllNSGA* |
| *espresso* | 7.24 | 1.4 | 1.4 | 1.5 | 1.5 |
| *gawk* | 3.43 | 3.2 | 6.7 | 4.4 | 4.4 |
| *flex* | 0.13 | 7.9 | 10.0 | 6.2 | 11.6 |
| *sed* | 0.25 | 9.4 | 7.0 | 7.0 | 5.4 |

| Memory Original (Peak/Wasted KB) | Wasted Memory Reduction (%) | | | |
|---|---|---|---|---|
| | *ShaRand* | *ShaNSGA* | *AllRand* | *AllNSGA* |
| 3500/521 | 6.1 | 6.1 | 0 | 19.2 |
| 29680/3552 | 15.6 | 15.6 | 16.2 | 20.9 |
| 10816/525 | 13.0 | 13.0 | 0 | 12.2 |
| 7048/948 | 3.8 | 3.8 | 2.1 | 17.9 |

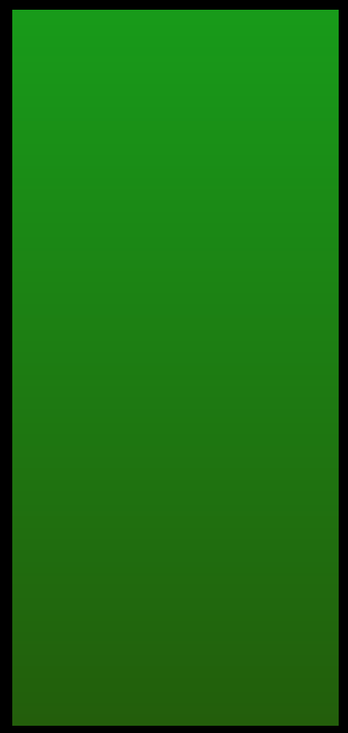
# Results : Execution cost

**Table 6: Computation Cost in Time**

| Subject | Optimisation Time (h) | | | | Exposing Time (h) | Extra Time Needed for **NSGA* (%) |
|---|---|---|---|---|---|---|
| | *ShaRand* | *ShaNSGA* | *AllRand* | *AllNSGA* | | |
| *espresso* | 39.7 | 46.4 | 9.0 | 39.3 | 12.5 | 18.5 |
| *gawk* | 22.7 | 18.4 | 13.9 | 16.4 | 5.4 | 11.7 |
| *flex* | 7.7 | 6.3 | 5.3 | 5.0 | 1.3 | 0.7 |
| *sed* | 9.4 | 7.6 | 5.9 | 6.6 | 1.9 | 12.6 |

# Deep Parameter Example

Heap



Managed by allocator

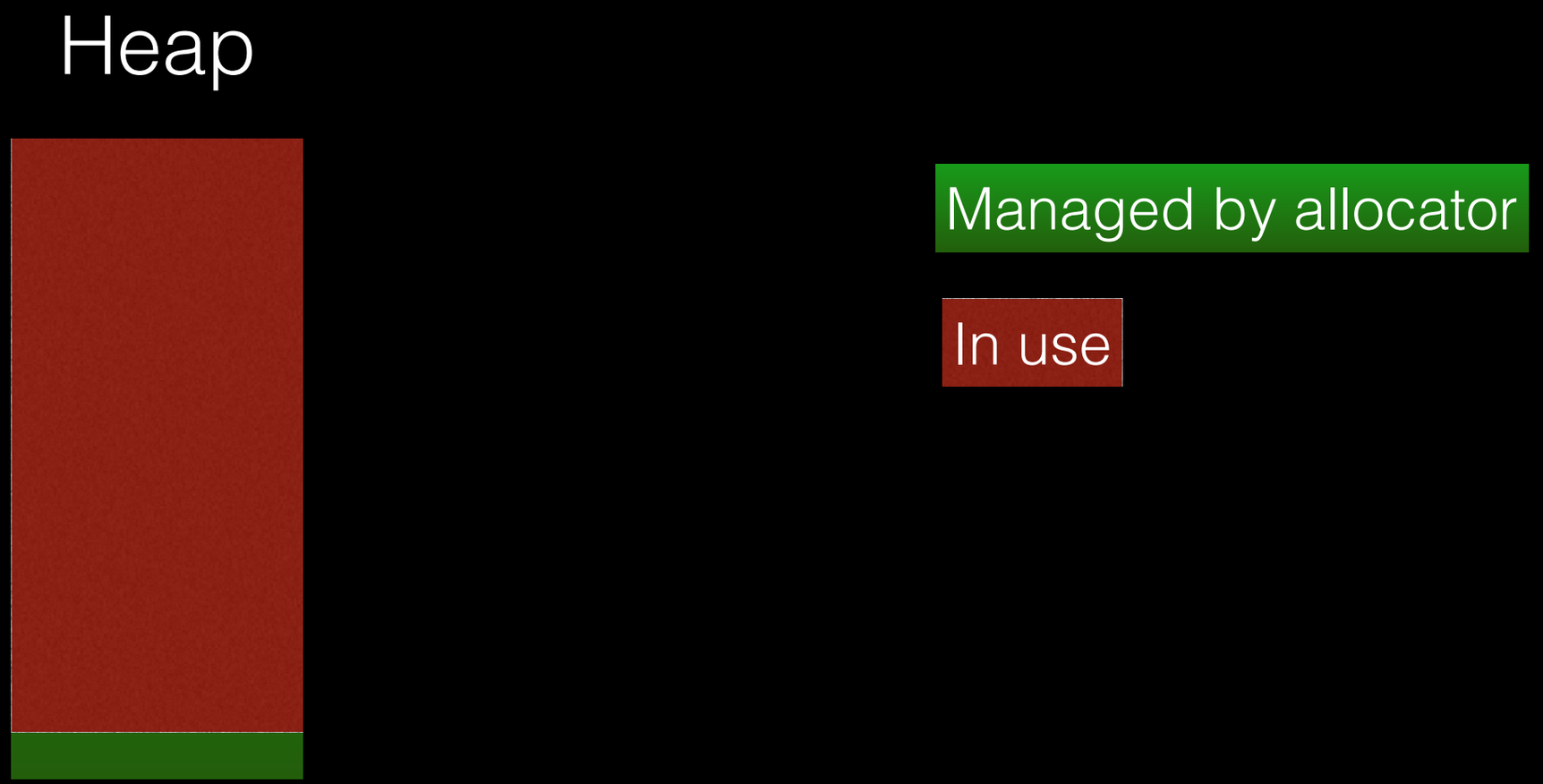
# Deep Parameter Example

Heap

Managed by allocator

In use

# Deep Parameter Example



# Deep Parameter Example

Heap

Managed by allocator

In use

# Deep Parameter Example

Heap

Managed by allocator

In use

```
1 static void* sys_alloc(mstate m,size_t nb)
2 {
3 ...
4 /* Subtract out existing available top
space from MORECORE request. */
5 ssize = granularity_align(nb - m->topsize
+ SYS_ALLOC_PADDING+EXPOSE_4334);
6 ...
7 }
```

# Related work

Software Tuning panel for Autonomic Control (STAC)

Autotuner / PowerDial

# Summary

Mutation analysis based approach to expose deep parameters

12% on time, 21% on memory