Advancing Software Analysis via Changed Perspectives

Zhendong Su

ETH Zurich



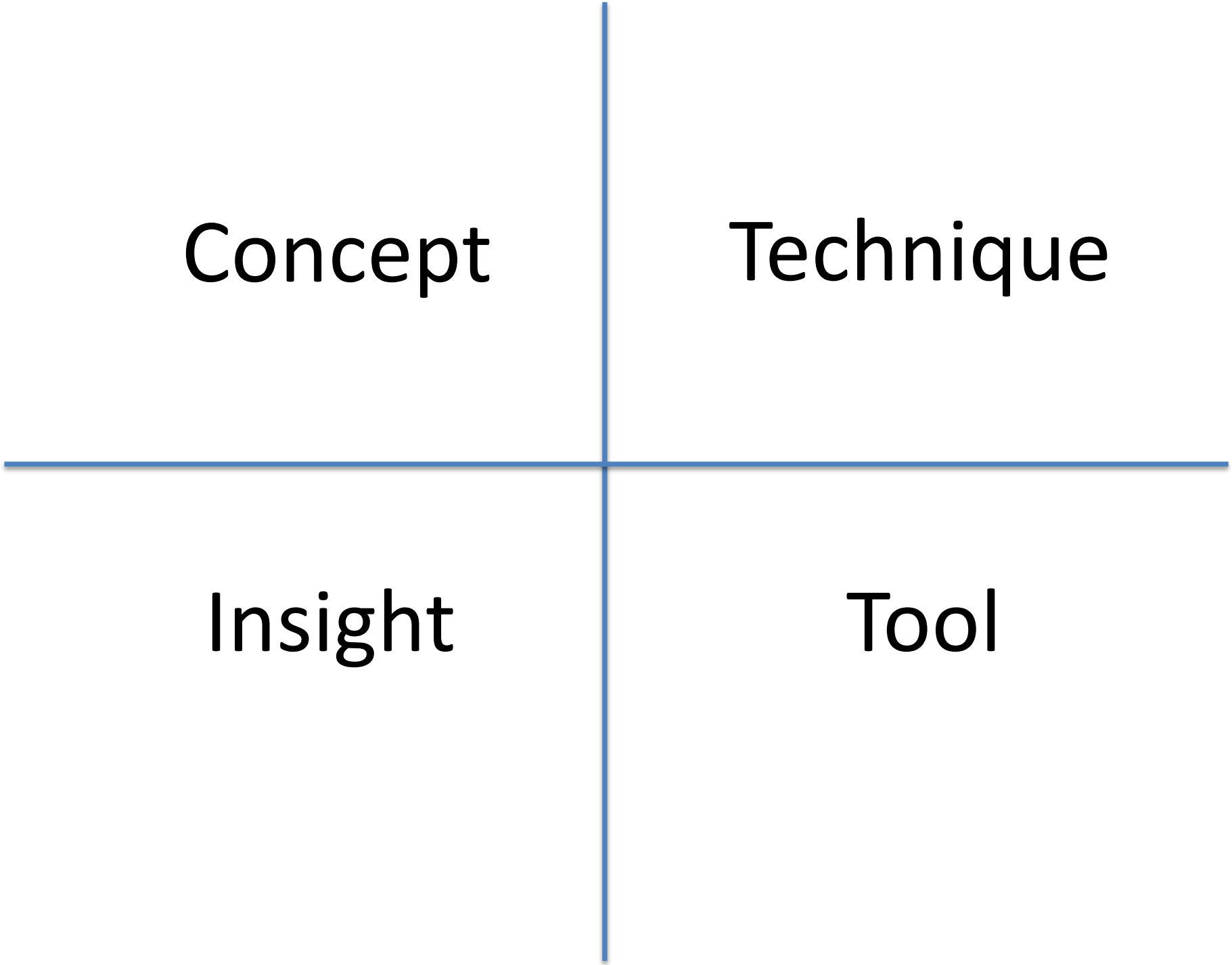


Mehrdad Afshari **Zhoulai Fu** **Vu Le** **Chengnian Sun** **Qirun Zhang**


PERSPECTIVE
IS
EVERYTHING


1/2
1/2

What is research impact?

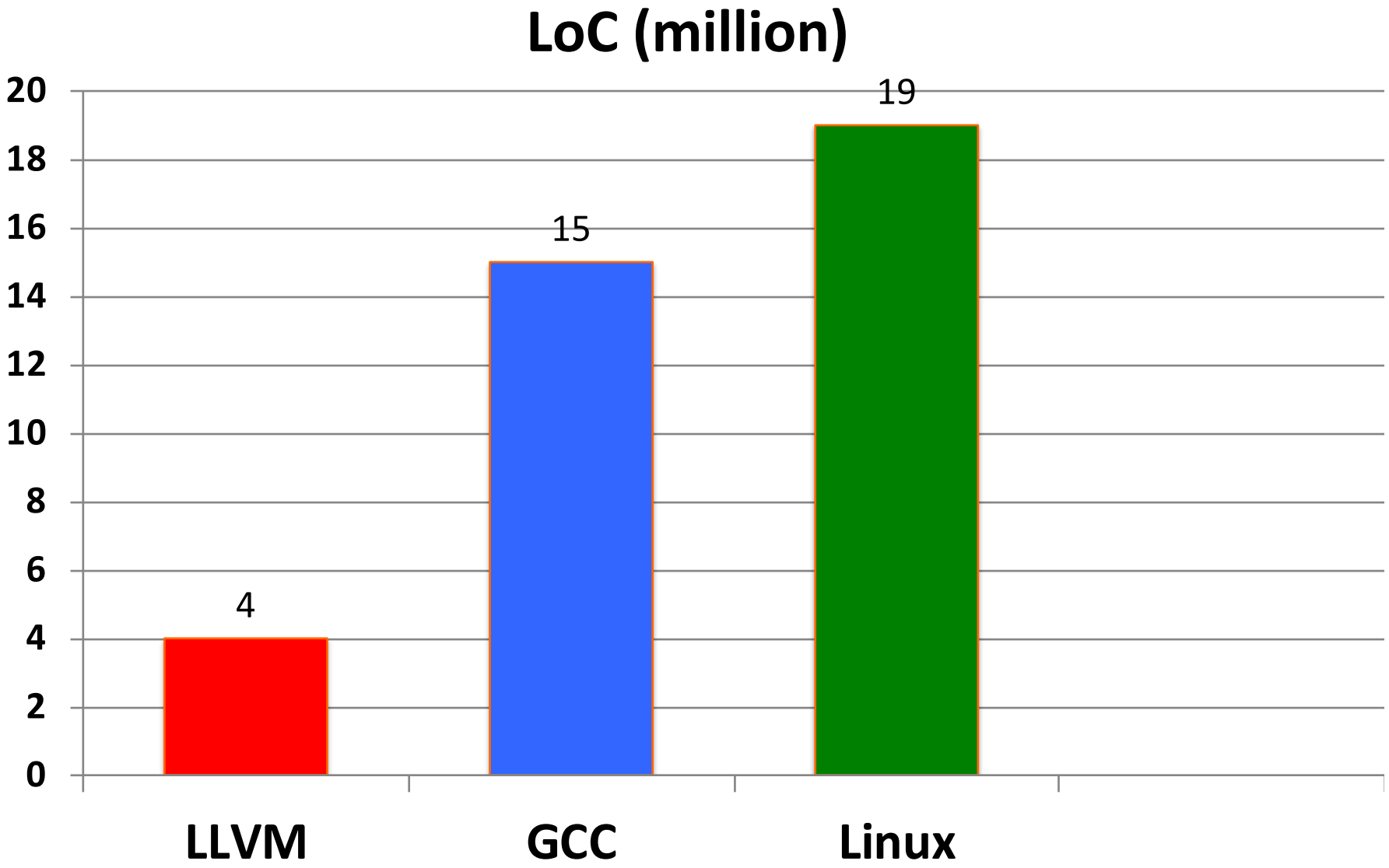

Concept
Technique
Insight
Tool

Two instances

- **Validate production compilers**
  - Black-box analysis
- **Analyze floating-point software**
  - Dynamic analysis

Validate Production Compilers

Compiler complexity



LLVM bug 14972

```
struct tiny { char c; char d; char e; };

void foo(struct tiny x) {
    if (x.c != 1) abort();
    if (x.e != 1) abort();
}

int main() {
    struct tiny s;
    s.c = 1; s.d = 1; s.e = 1;
    foo(s);
    return 0;
}
```

```
$ clang -m32 -O0 test.c ; ./a.out
$ clang -m32 -O1 test.c ; ./a.out
Aborted (core dumped)
```

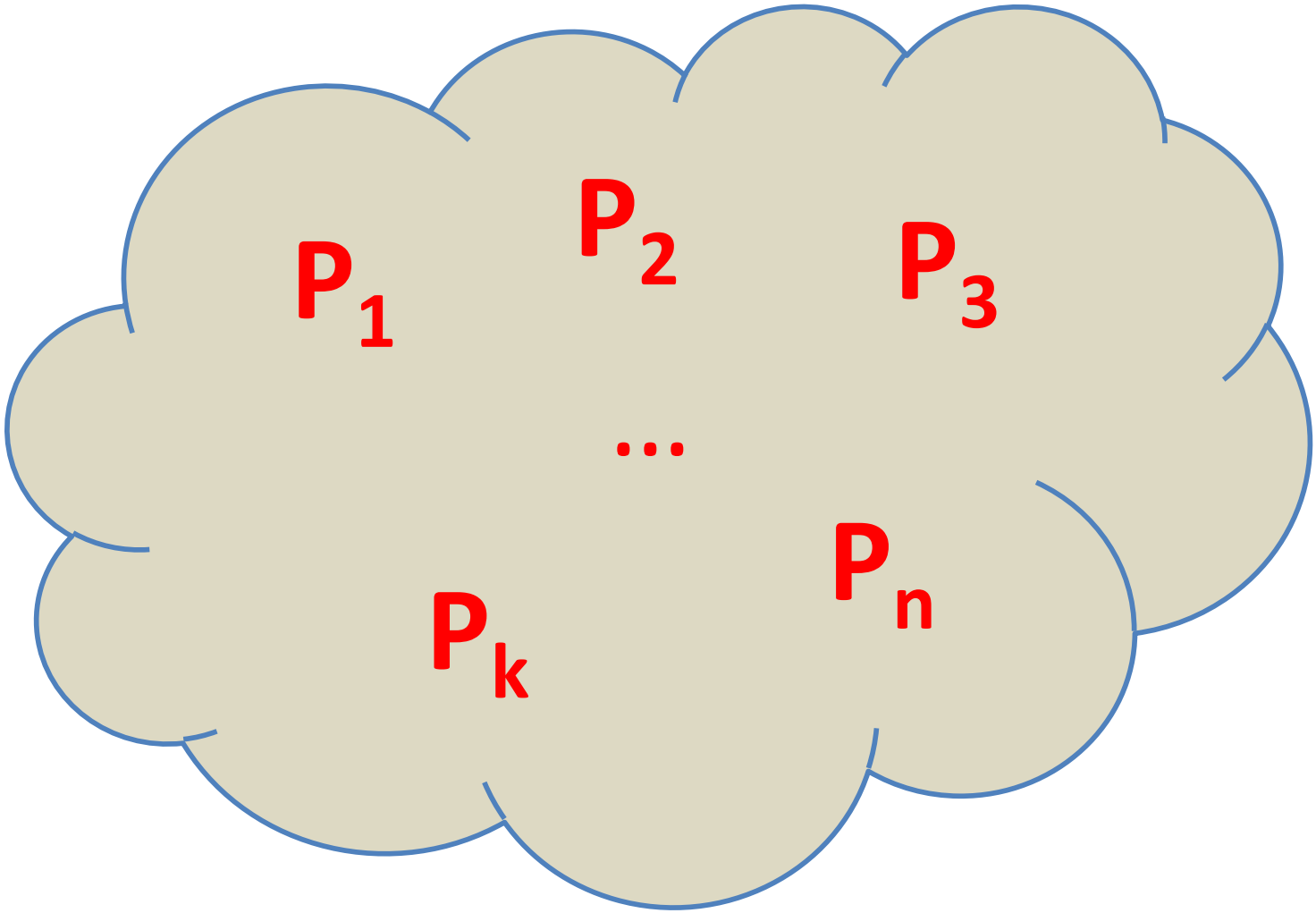
Developer comment

"... very, very concerning when I got to the root cause, and very annoying to fix ..."

http://llvm.org/bugs/show_bug.cgi?id=14972

Vision

$P \equiv$



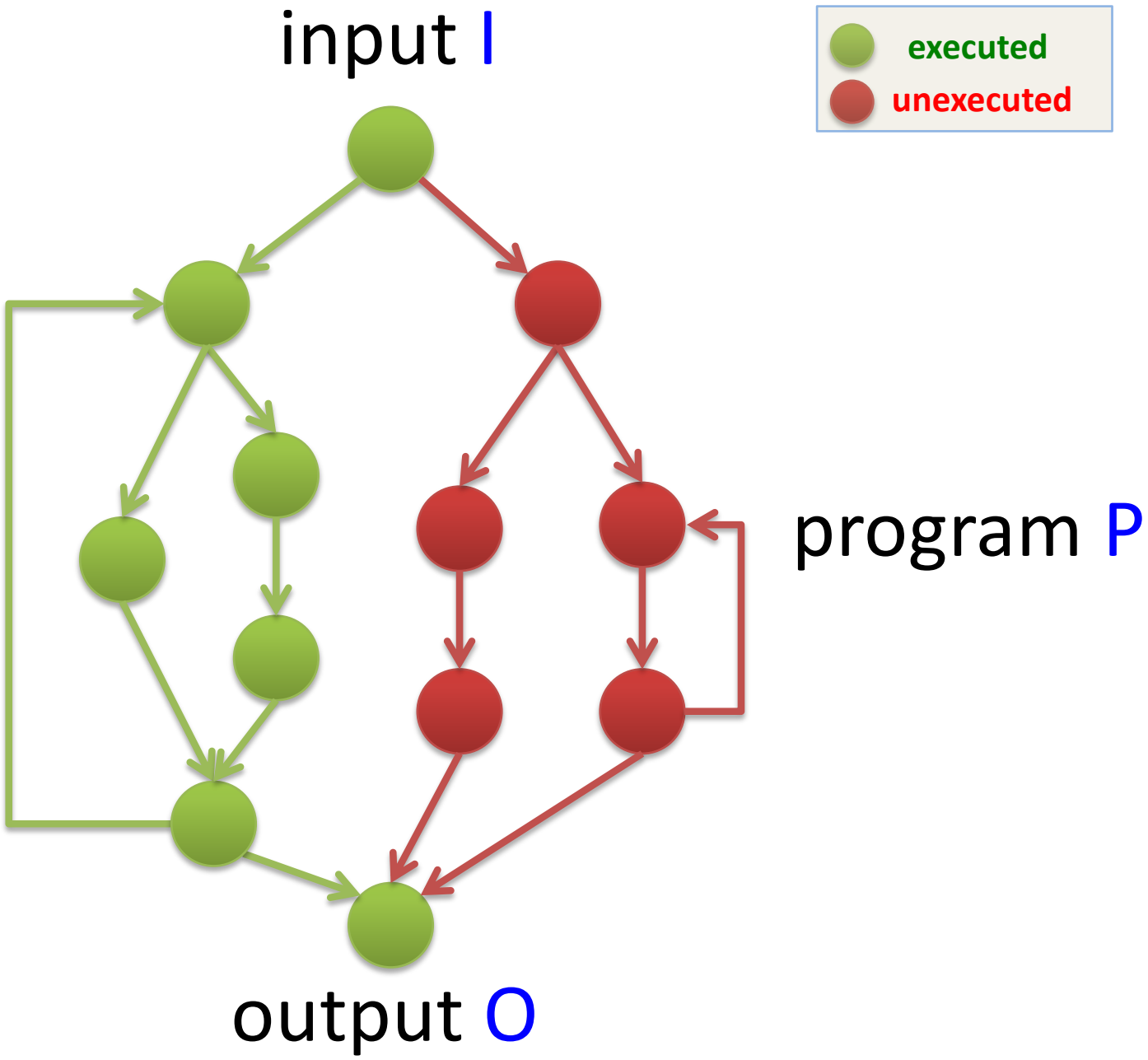
Key challenges

- Generation
  - How to generate **different**, yet **equivalent** tests?
- Validation
  - How to check that tests are **indeed equivalent**?
- Both are long-standing hard issues

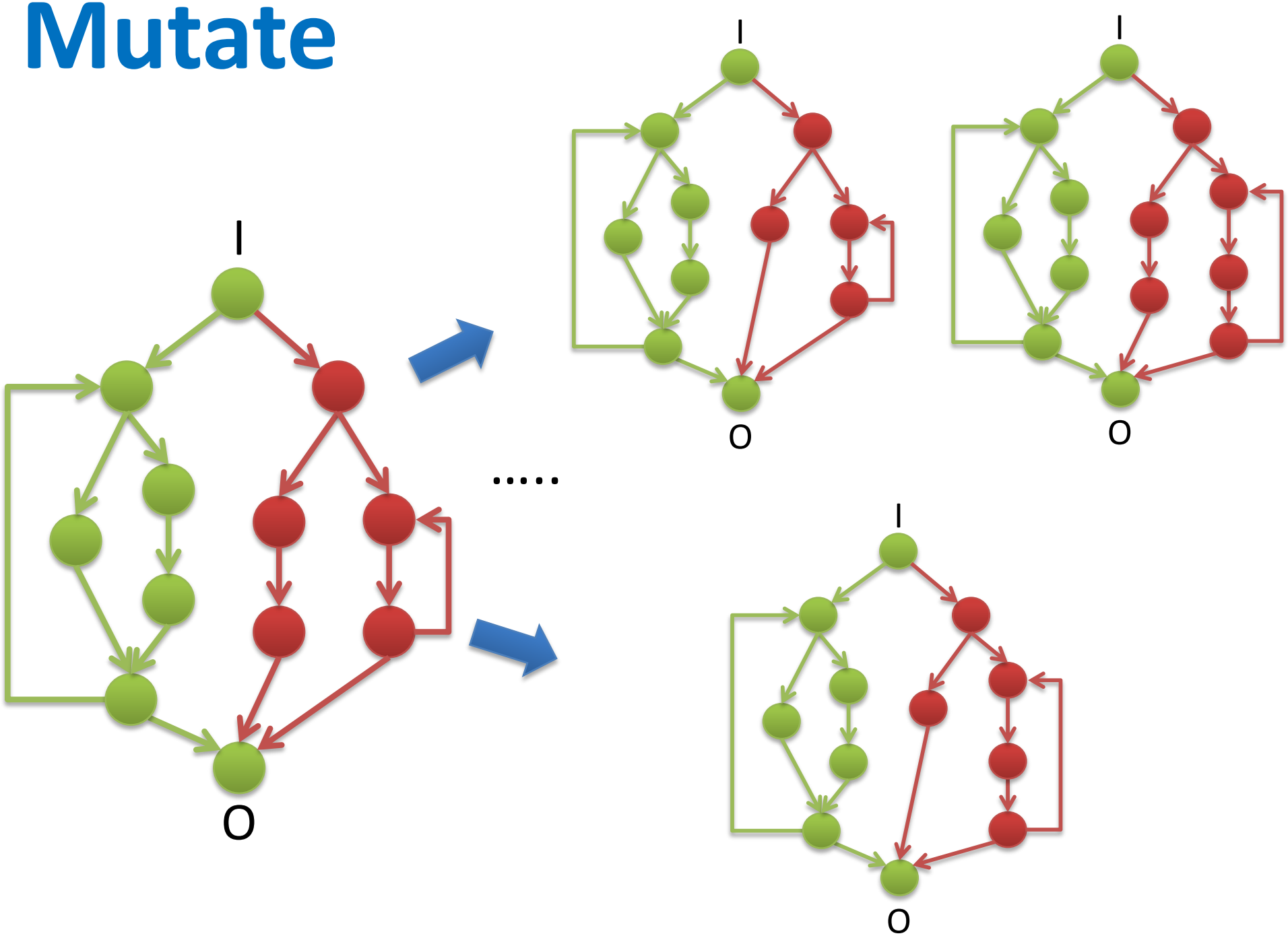
Equiv. modulo inputs

- Relax equiv. wrt a **given input i**
  - **Must**: **$P(i) = P_k(i)$** on input **i**
  - **Okay**: **$P(j) \neq P_k(j)$** on all input **$j \neq i$**
- Exploit close **interplay** between
  - **Dynamic** program execution on **some input**
  - **Static** compilation for **all input**

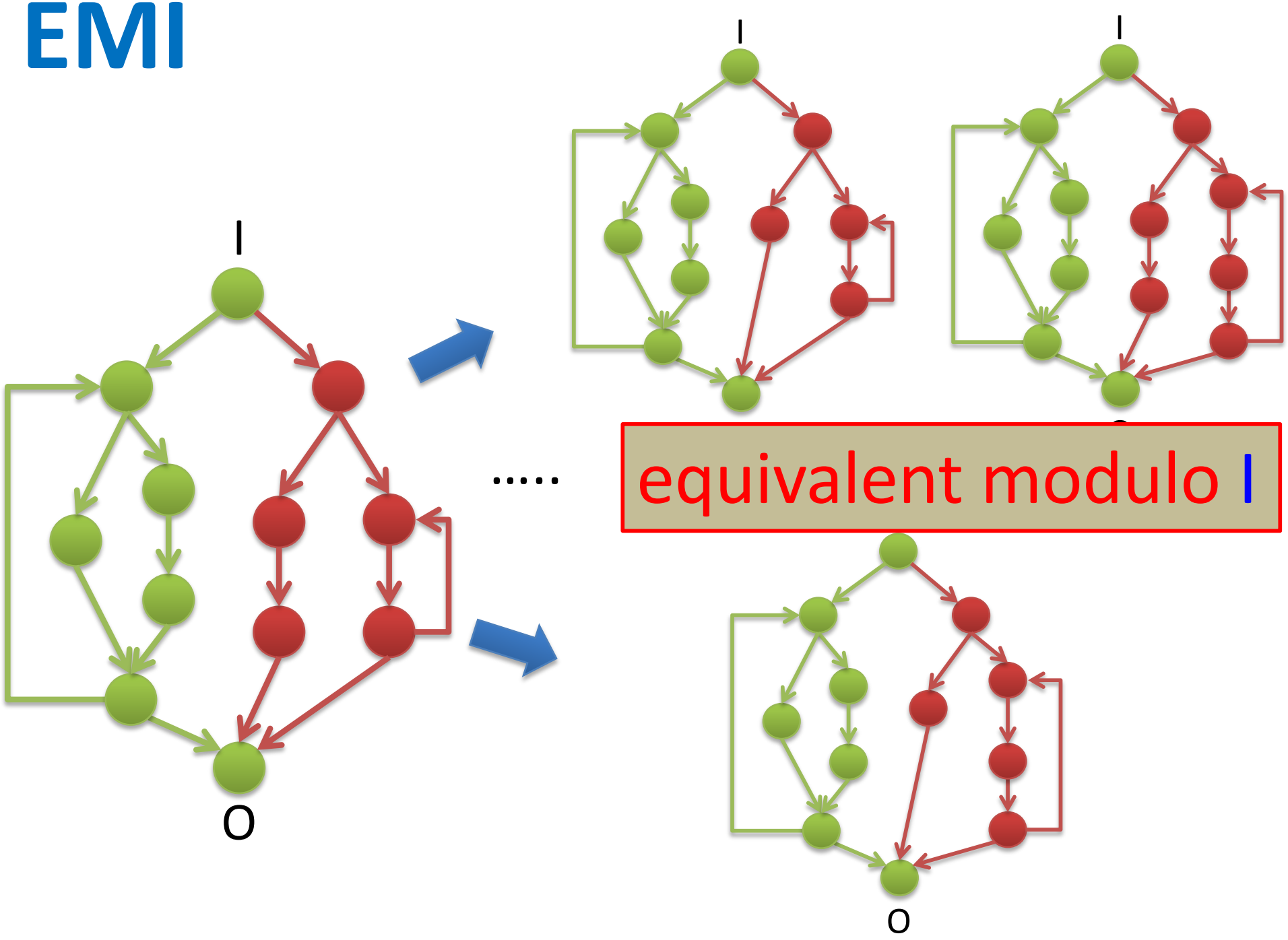
Profile



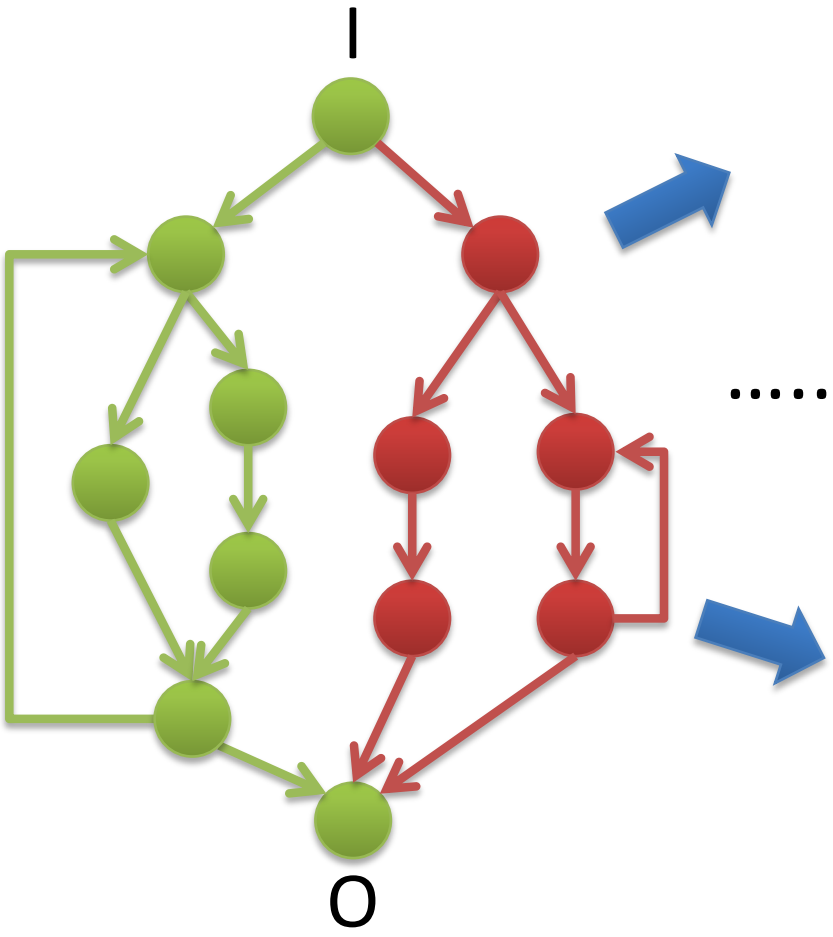
Mutate



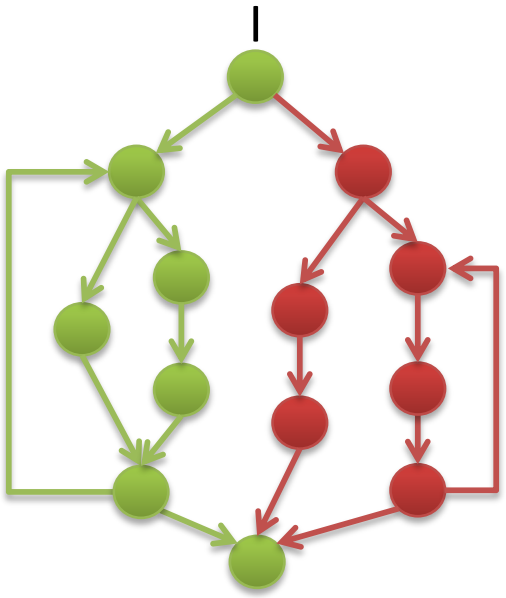
EMI



Find bugs



.....



O’ ≠ O

Revisit challenges

- Generation (**easy**)
  - How to generate **different**, yet **equivalent** tests?
- Validation (**easy**)
  - How to check that tests are **indeed equivalent**?
- ~~Both are long-standing hard issues~~

LLVM bug 14972

```
struct tiny { char c; char d; char e; };

void foo(struct tiny x) {
    if (x.c != 1) abort();
    if (x.e != 1) abort();
}

int main() {
    struct tiny s;
    s.c = 1; s.d = 1; s.e = 1;
    foo(s);
    return 0;
}
```

```
$ clang -m32 -O0 test.c ; ./a.out
$ clang -m32 -O1 test.c ; ./a.out
Aborted (core dumped)
```

Seed file

```
struct tiny { char c; char d; char e; };
f(int n, struct tiny x, struct tiny y,
  struct tiny z, long l) {
    if (x.c != 10) abort();
    if (x.d != 20) abort();
    if (x.e != 30) abort();
    if (y.c != 11) abort();
    if (y.d != 21) abort();
    if (y.e != 31) abort();
    if (z.c != 12) abort();
    if (z.d != 22) abort();
    if (z.e != 32) abort();
    if (l != 123) abort();
}
main() {
    struct tiny x[3];
    x[0].c = 10;
    x[1].c = 11;
    x[2].c = 12;
    x[0].d = 20;
    x[1].d = 21;
    x[2].d = 22;
    x[0].e = 30;
    x[1].e = 31;
    x[2].e = 32;
    f(3, x[0], x[1], x[2], (long)123);
    exit(0);
}
```

```
$ clang -m32 -O0 test.c ; ./a.out
$ clang -m32 -O1 test.c ; ./a.out
```

Seed file

```
struct tiny { char c; char d; char e; };
f(int n, struct tiny x, struct tiny y,
  struct tiny z, long l) {
    if (x.c != 10) abort();
    if (x.d != 20) abort();
    if (x.e != 30) abort();
    if (y.c != 11) abort();
    if (y.d != 21) abort();
    if (y.e != 31) abort();
    if (z.c != 12) abort();
    if (z.d != 22) abort();
    if (z.e != 32) abort();
    if (l != 123) abort();
}
main() {
    struct tiny x[3];
    x[0].c = 10;
    x[1].c = 11;
    x[2].c = 12;
    x[0].d = 20;
    x[1].d = 21;
    x[2].d = 22;
    x[0].e = 30;
    x[1].e = 31;
    x[2].e = 32;
    f(3, x[0], x[1], x[2], (long)123);
    exit(0);
}
```

unexecuted

```
$ clang -m32 -O0 test.c ; ./a.out
$ clang -m32 -O1 test.c ; ./a.out
```

Transformed file

```
struct tiny { char c; char d; char e; };
f(int n, struct tiny x, struct tiny y,
  struct tiny z, long l) {
    if (x.c != 10) /* deleted */;
    if (x.d != 20) abort();
    if (x.e != 30) /* deleted */;
    if (y.c != 11) abort();
    if (y.d != 21) abort();
    if (y.e != 31) /* deleted */;
    if (z.c != 12) abort();
    if (z.d != 22) /* deleted */;
    if (z.e != 32) abort();
    if (l != 123) /* deleted */;
}
main() {
    struct tiny x[3];
    x[0].c = 10;
    x[1].c = 11;
    x[2].c = 12;
    x[0].d = 20;
    x[1].d = 21;
    x[2].d = 22;
    x[0].e = 30;
    x[1].e = 31;
    x[2].e = 32;
    f(3, x[0], x[1], x[2], (long)123);
    exit(0);
}
```

```
$ clang -m32 -O0 test.c ; ./a.out
$ clang -m32 -O1 test.c ; ./a.out
Aborted (core dumped)
```

Reduced file

```
struct tiny { char c; char d; char e; };

void foo(struct tiny x) {
    if (x.c != 1) abort();
    if (x.e != 1) abort();
}

int main() {
    struct tiny s;
    s.c = 1; s.d = 1; s.e = 1;
    foo(s);
    return 0;
}
```

```
$ clang -m32 -O0 test.c ; ./a.out
$ clang -m32 -O1 test.c ; ./a.out
Aborted (core dumped)
```

LLVM bug autopsy

```
struct tiny { char c; char d; char e; };

void foo(struct tiny x) {
    if (x.c != 1) abort();
    if (x.e != 1) abort();
}

int main() {
    struct tiny s;
    s.c = 1; s.d = 1; s.e = 1;
    foo(s);
    return 0;
}
```

GVN: load struct using 32-bit load

```
$ clang -m32 -O0 test.c ; ./a.out
$ clang -m32 -O1 test.c ; ./a.out
Aborted (core dumped)
```

LLVM bug autopsy

```
struct tiny { char c; char d; char e; };

void foo(struct tiny x) {
    if (x.c != 1) abort();
    if (x.e != 1) abort();
}

int main() {
    struct tiny s;
    s.c = 1; s.d = 1; s.e = 1;
    foo(s);
    return 0;
}
```

GVN: load struct using 32-bit load

SRoA: read past the struct's end → undefined behavior

```
$ clang -m32 -O0 test.c ; ./a.out
$ clang -m32 -O1 test.c ; ./a.out
Aborted (core dumped)
```

LLVM bug autopsy

```
struct tiny { char c; char d; char e; };

void foo(struct tiny x) {
    if (x.c != 1) abort();
    if (x.e != 1) abort();
}

int main() {
    struct tiny s;
    s.c = 1; s.d = 1; s.e = 1;
    foo(s);
    return 0;
}
```

GVN: load struct using 32-bit load

SRoA: read past the struct's end → undefined behavior

remove

```
$ clang -m32 -O0 test.c ; ./a.out
$ clang -m32 -O1 test.c ; ./a.out
Aborted (core dumped)
```

GCC bug 58731

```
int a, b, c, d, e;
int main() {
    for (b = 4; b > -30; b--)
        for (; c;)
            for (;;) {
                e = a > 2147483647 - b;
                if (d) break;
            }
    return 0;
}
```

```
$ gcc -O0 test.c ; ./a.out
$ gcc -O3 test.c ; ./a.out
^C
```

GCC bug autopsy

```
int a, b, c, d, e;
int main() {
    for (b = 4; b > -30; b--)
        for (; c;)
            for (;;) {
                e = a > 2147483647 - b;
                if (d) break;
            }
    return 0;
}
```

PRE: loop invariant

```
$ gcc -O0 test.c ; ./a.out
$ gcc -O3 test.c ; ./a.out
^C
```

GCC bug autopsy

```
int a, b, c, d, e;
int main() {
    for (b = 4; b > -30; b--)
        int f = 2147483647 - b;
        for (; c;)
            for (;;) {
                e = a > f;
                if (d) break;
            }
    return 0;
}
```

LIM

```
$ gcc -O0 test.c ; ./a.out
$ gcc -O3 test.c ; ./a.out
^C
```

GCC bug autopsy

```
int a, b, c, d, e;
int main() {
    for (b = 4; b > -30; b--)
        int f = 2147483647 - b;
        for (; c;)
            for (;;) {
                e = a > f;
                if (d) break;
            }
    return 0;
}
```

integer overflow

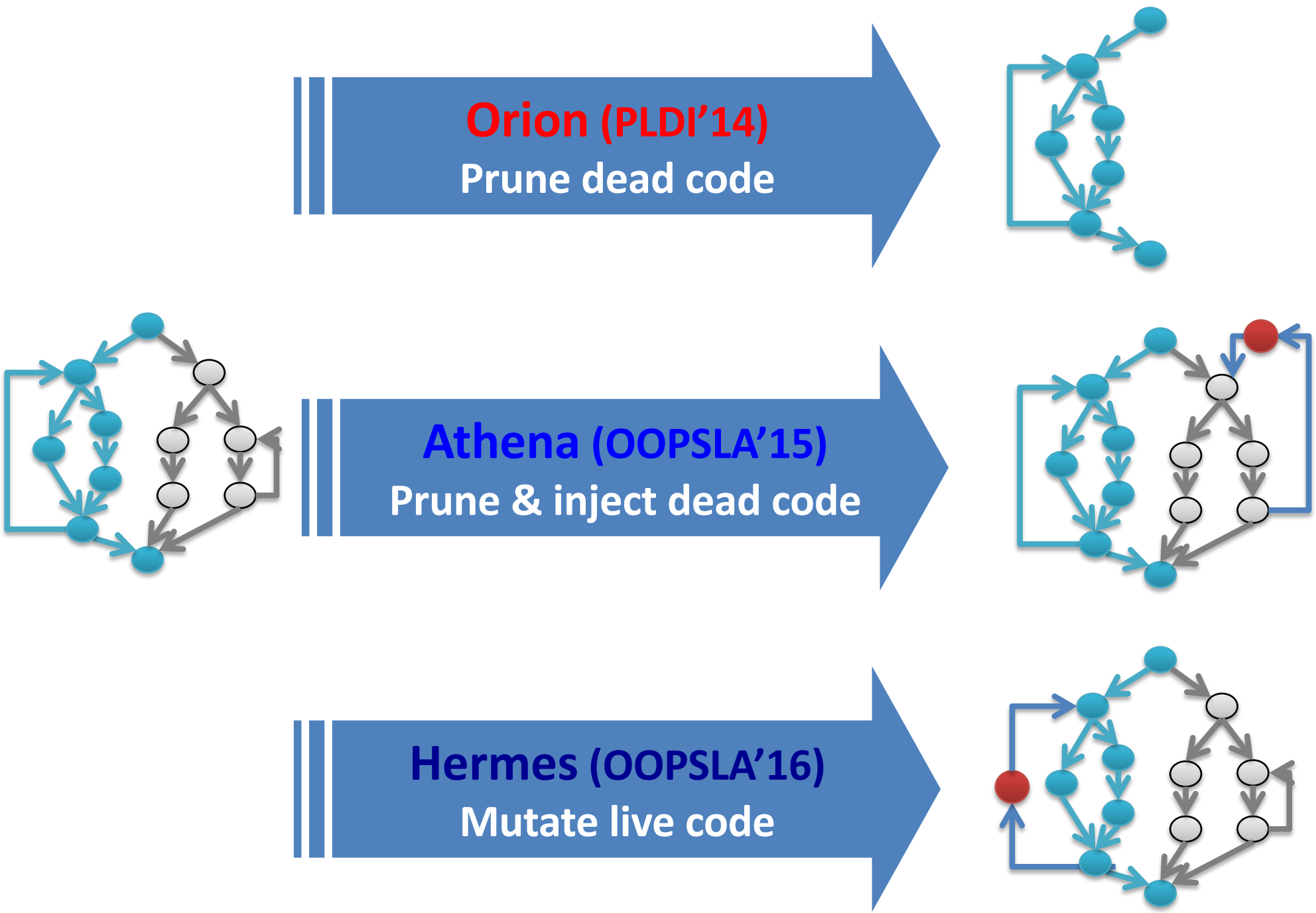
```
$ gcc -O0 test.c ; ./a.out
$ gcc -O3 test.c ; ./a.out
^C
```

Seed program

```
int a, b, c, d, e;
int main() {
    for (b = 4; b > -30; b--)
        for (; c;)
            for (;;) {
                b++;
                e = a > 2147483647 - b;
                if (d) break;
            }
    return 0;
}
```

no longer a loop invariant

```
$ gcc -O0 test.c ; ./a.out
$ gcc -O3 test.c ; ./a.out
```


Orion (PLDI'14)
Prune dead code
Athena (OOPSLA'15)
Prune & inject dead code
Hermes (OOPSLA'16)
Mutate live code

bug counts

	GCC	LLVM	TOTAL
Reported	841	781	**1622**
Fixed	612	419	**1031**

- **ISSTA'15: Stress-testing link-time optimization**
- **ICSE'16: Analyzing compilers' diagnostic support**
- **PLDI'17: Skeletal program enumeration (SPE)**

LLVM 3.9 & 4.0 Release Notes

"… thanks to **Zhendong Su and his team**
whose fuzz testing **prevented many bugs**
going into the release …"

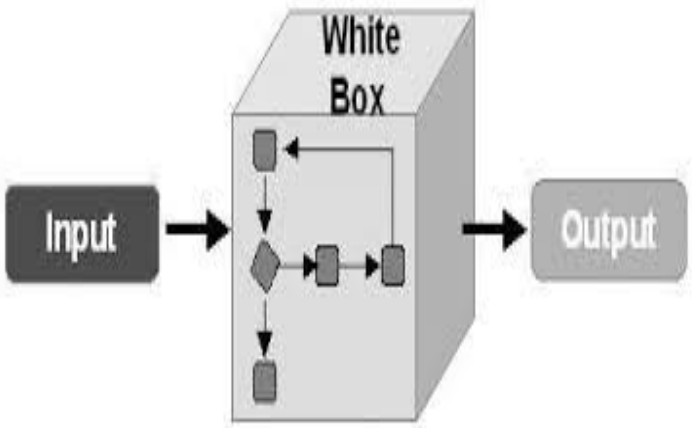
GCC's list of contributors

https://gcc.gnu.org/onlinedocs//gcc/Contributors.html

"Zhendong Su … for reporting numerous bugs"
"Chengnian Sun … for reporting numerous bugs"
"Qirun Zhang … for reporting numerous bugs"

***Take-away**: Not only do good research, but be its **loyal**, **continuous** user!*

Analyze Floating-Point Software



Floating-point code



- **Important**: bugs can lead to disasters
- **Challenging**: hard to get right

Why difficult?

- FP Math ≠ Real Math
- Non-linear relations
- Transcendental functions

  sin, log, exp, …

```
1  double foo(double x){
2    if (x<=1.0)
3      x++;
4
5    y = x*x;
6    if (y<=4.0)
7      x--;
8    return x;
9  }
```

Challenging for all known approaches

New perspective: ME

(p, ϕ)

Analyzing numerical programs

- Coverage-based testing
- Boundary value analysis
- Numerical exception detection

Floating-point constraint solving

Mathematical Execution (ME)

r

Mathematical optimization (MO)

input **x** drives **p** to satisfy ϕ $\leftrightarrow$ **x** minimizes **r**

FP constraints

Solving the floating-point constraint π

$$(SIN(x) == x) \wedge (x \geq 10^{-10})$$

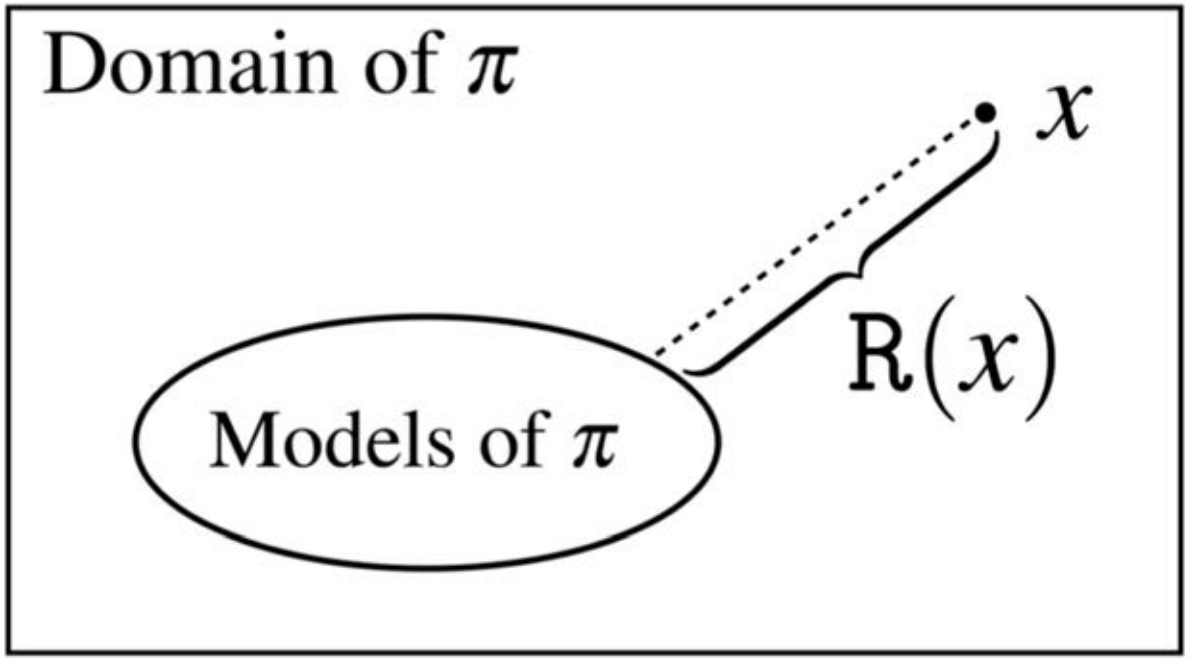
- Satisfiable if x is floating-point

$$\text{For } x \in \mathbb{F}, SIN(x) = x \Leftrightarrow x \simeq 0$$

- Unsatisfiable if x is real

$$\text{For } x \in \mathbb{R}, SIN(x) = x \Leftrightarrow x = 0$$

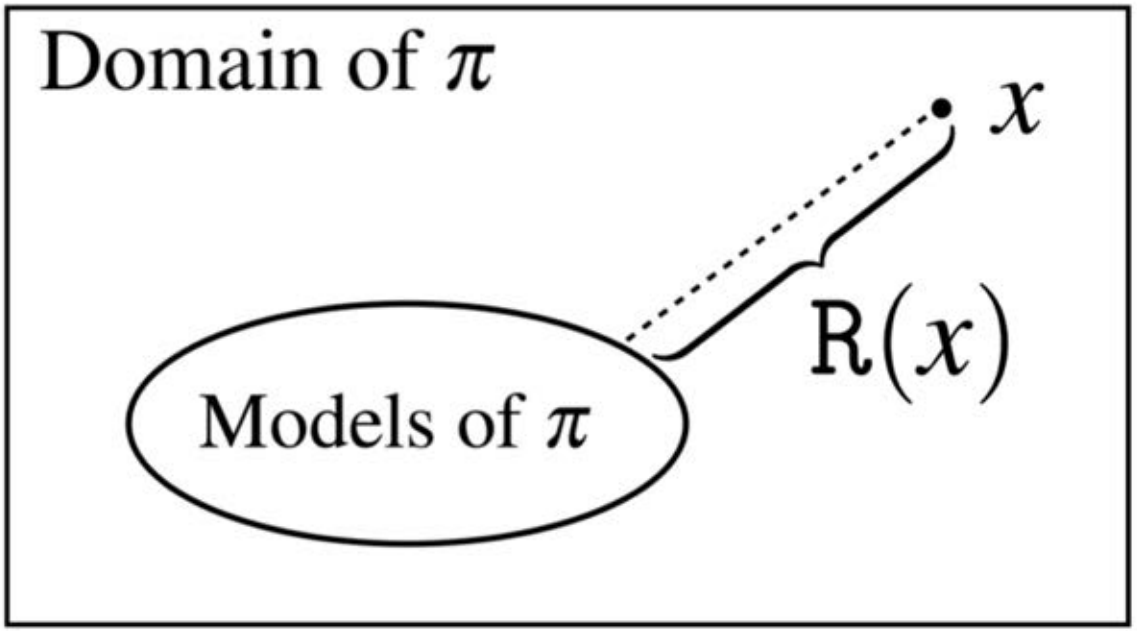
Step 1



Simulate π with a floating-point program R

- $\mathrm{R}(x) \geq 0$ for all x
- $\mathrm{R}(x) = 0 \Leftrightarrow x \models \pi$

Step 2



Minimize R as if it is a mathematical function

- Let x^* be the minimum point

$$\pi \text{ satisfiable} \Leftrightarrow \mathsf{R}(x^*) = 0$$

Construct R

Necessary Conditions to meet :

1. $\mathsf{R}(x) \geq 0$ for all x
2. $\mathsf{R}(x) = 0 \Leftrightarrow x \models \pi$

How?

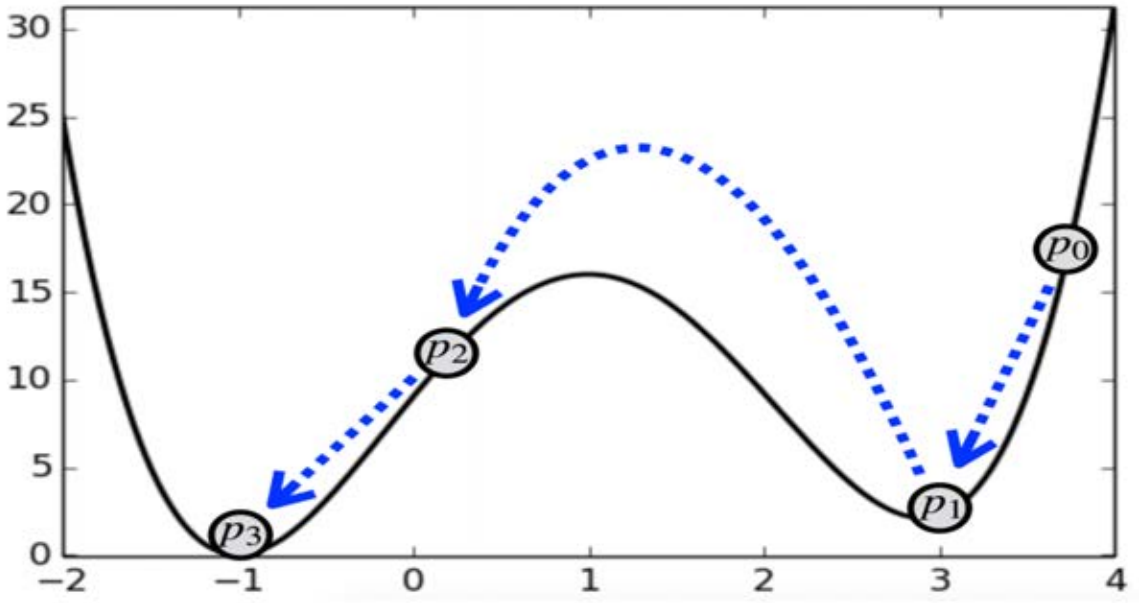
Constraint π	Program R
$x == y$	$(x-y)^2$
$x \leq y$	$x \leq y \ ? \ 0 : (x-y)^2$
$\pi_1 \wedge \pi_2$	$\mathsf{R}_1 + \mathsf{R}_2$
$\pi_1 \vee \pi_2$	$\mathsf{R}_1 * \mathsf{R}_2$

R can be constructed from a CNF form

Minimize R

Unconstrained programming techniques:

- Local optimization
- Monte Carlo Markov Chain (MCMC)
- We use them as black-box
- Do not analyze π; execute R



Theoretical guarantees

Let R satisfy (1) $\mathsf{R}(x) \geq 0$, and (2) $\mathsf{R}(x) = 0 \Leftrightarrow x \models \pi$, and x^* be a minimum point of R. Then

$$\pi \text{ satisfiable} \Leftrightarrow \mathsf{R}(x^*) = 0.$$

Threats

- Floating-point inaccuracy when calculating with R
- Sub-optimal x^*

Example

$$(SIN(x) == x) \wedge (x \geq 10^{-10})$$

$$\Downarrow$$

$$(SIN(x) - x)^2 + \begin{cases} 0 & \text{if } x \geq 10^{-10} \\ (x - 10^{-10})^2 & \text{otherwise} \end{cases}$$

$$\Downarrow$$

$$x^* = 9.0 * 10^{-9} \text{ (can be others)}$$

XSat & results

- Developed the ME-based XSat tool
- Evaluated against MathSat and Z3
- Used SMT-Comp 2015 FP benchmarks
- Result summary
  - 100% consistent results
  - 700+X faster than MathSat
  - 800+X faster than Z3

***Take-away**: Don't be afraid of difficult problems, look at them from **new perspectives**,*

even for "damned" problems!

Generalizations

- Coverage-based testing of FP code
- Boundary value analysis
- FP exception detection
- Path divergence detection

Coverage-based testing

Goal

To generate test inputs to cover all branches of a program like this:

- pointer operations: &, *
- type casting: (int*), (unsigned)
- bit operations ^, &, >>
- floating-point comparison

```
double __ieee754_fmod(double x, double y){
  ...
  Zero[] = {0.0, -0.0,};
  hx = *(1+(int*)&x);
  lx = *(int*)&x;
  hy = *(1+(int*)&y);
  ly = *(int*)&y;
  sx = hx&0x80000000;
  hx ^=sx;
  hy &= 0x7fffffff;

  if((hy|ly)==0||(hx>=0x7ff00000)||
     ((hy|((ly|-ly)>>31))>0x7ff00000))
    return (x*y)/(x*y);
  if(hx<=hy) {
    if((hx<hy)||(lx<ly)) return x;
    if(lx==ly)
      return Zero[(unsigned)sx>>31];
}


  if(hx<0x00100000) {
   if(hx==0) {
```

State-of-the-art & Challenges

Symbolic execution

- Path explosion
- Constraint solving

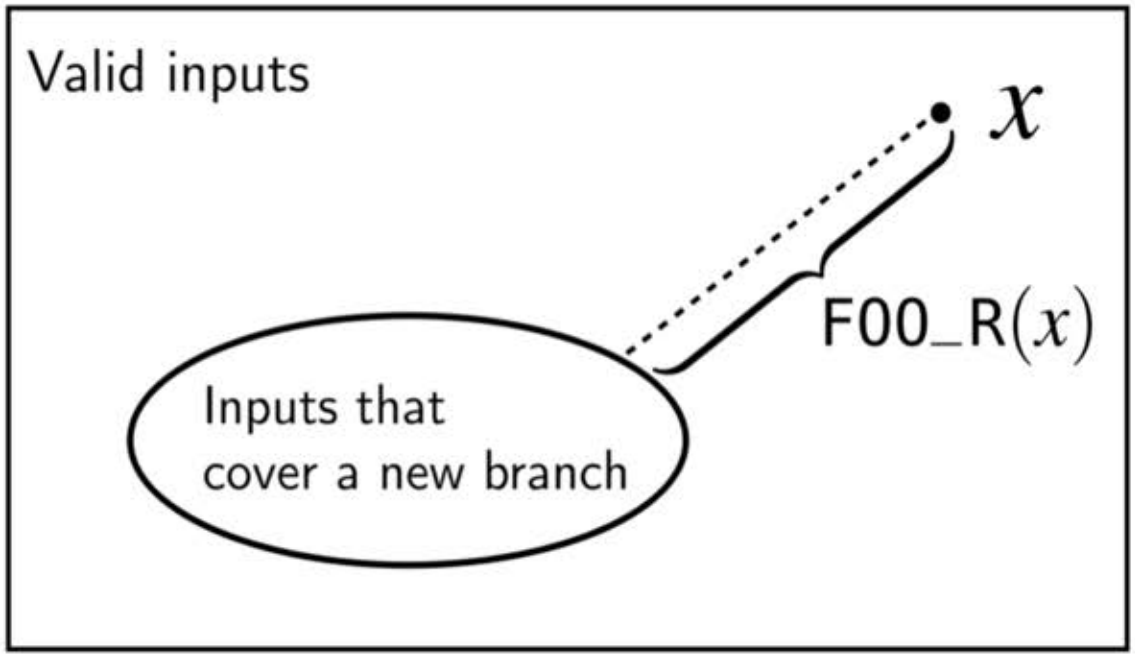
Search-based testing

- Fitness function
- Search strategies

Our approach

- No path issues
- No need to solve constraints
- Effective for FP programs

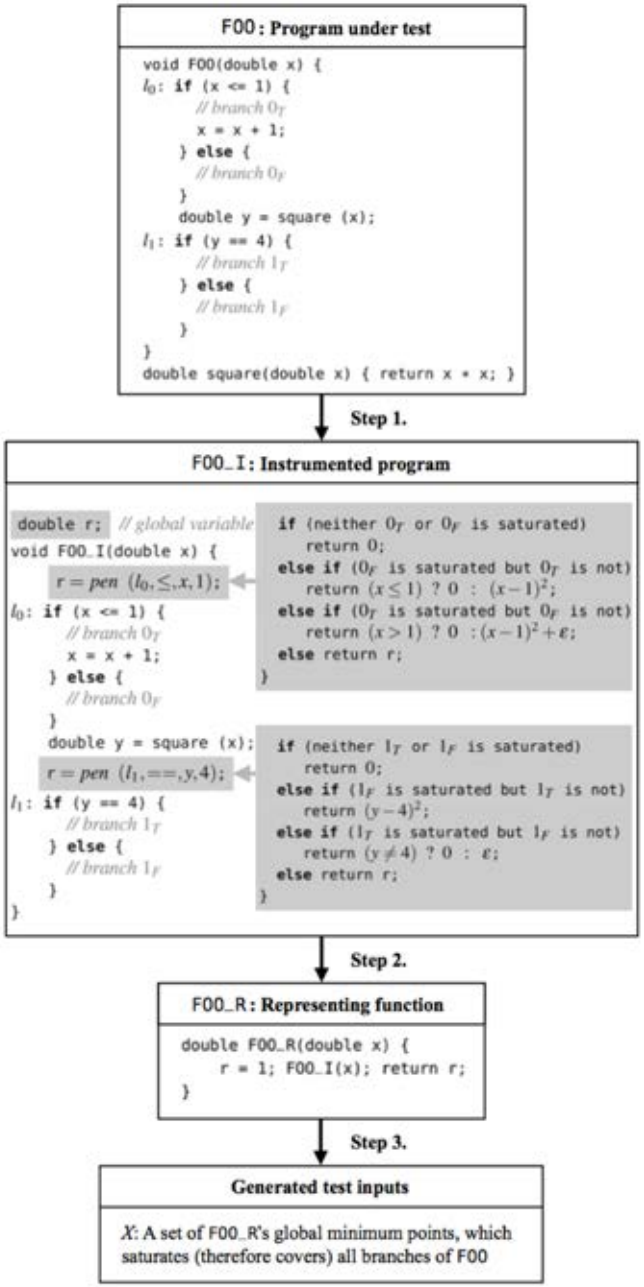
Our approach



Step 1: Derive a program FOO_R from FOO *s.t.*

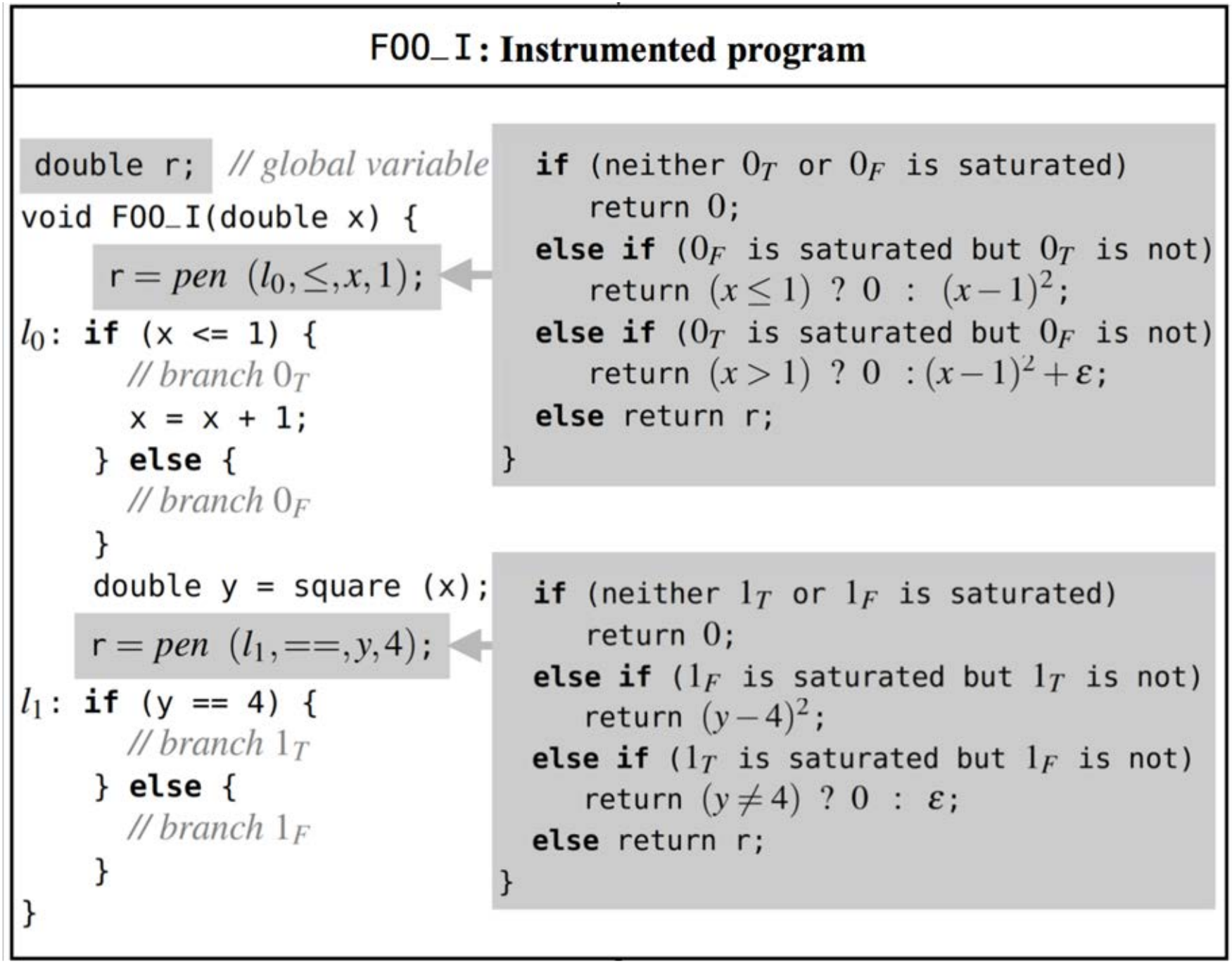
- FOO_R(x) ≥ 0 for all x, and
- FOO_R(x) $= 0 \Leftrightarrow x$ covers a new branch

Step 2: Repeatedly minimize FOO_R until > 0



FOO: Program under test

```
void FOO(double x) {
l0: if (x <= 1) {
      // branch 0T
      x = x + 1;
    } else {
      // branch 0F
    }
    double y = square (x);
l1: if (y == 4) {
      // branch 1T
    } else {
      // branch 1F
    }
}
double square(double x) { return x * x; }
```


FOO_I : Instrumented program
double r; // global variable
void FOO_I(double x) {
r = pen (l0, ≤, x, 1);
l0: if (x <= 1) {
// branch 0T
x = x + 1;
} else {
// branch 0F
}
double y = square (x);
r = pen (l1, ==, y, 4);
l1: if (y == 4) {
// branch 1T
} else {
// branch 1F
}
}
if (neither 0T or 0F is saturated)
return 0;
else if (0F is saturated but 0T is not)
return (x ≤ 1) ? 0 : (x − 1)^2;
else if (0T is saturated but 0F is not)
return (x > 1) ? 0 : (x − 1)^2 + ε;
else return r;
}
if (neither 1T or 1F is saturated)
return 0;
else if (1F is saturated but 1T is not)
return (y − 4)^2;
else if (1T is saturated but 1F is not)
return (y ≠ 4) ? 0 : ε;
else return r;
}

FOO_I: Instrumented program

```
double r;  // global variable
void FOO_I(double x) {
    r = pen (l0, ≤, x, 1);
l0: if (x <= 1) {
      // branch 0T
      x = x + 1;
    } else {
      // branch 0F
    }
    double y = square (x);
    r = pen (l1, ==, y, 4);
l1: if (y == 4) {
      // branch 1T
    } else {
      // branch 1F
    }
}
```

```
if (neither 0T or 0F is saturated)
    return 0;
else if (0F is saturated but 0T is not)
    return (x ≤ 1) ? 0 : (x−1)^2;
else if (0T is saturated but 0F is not)
    return (x > 1) ? 0 : (x−1)^2 + ε;
else return r;
}
```

```
if (neither 1T or 1F is saturated)
    return 0;
else if (1F is saturated but 1T is not)
    return (y−4)^2;
else if (1T is saturated but 1F is not)
    return (y ≠ 4) ? 0 : ε;
else return r;
}
```

FOO_R: Representing function

```
double FOO_R(double x) {
    r = 1; FOO_I(x); return r;
}
```

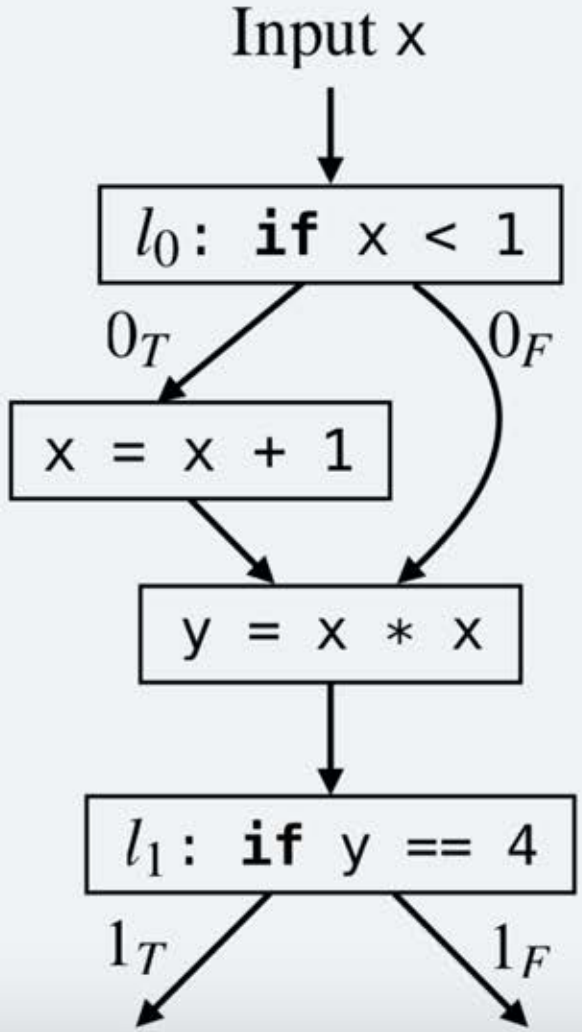
Generated test inputs

X: A set of FOO_R's global minimum points, which saturates (therefore covers) all branches of FOO

Example

Generate an input set to cover $\{0_T, 0_F, 1_T, 1_F\}$

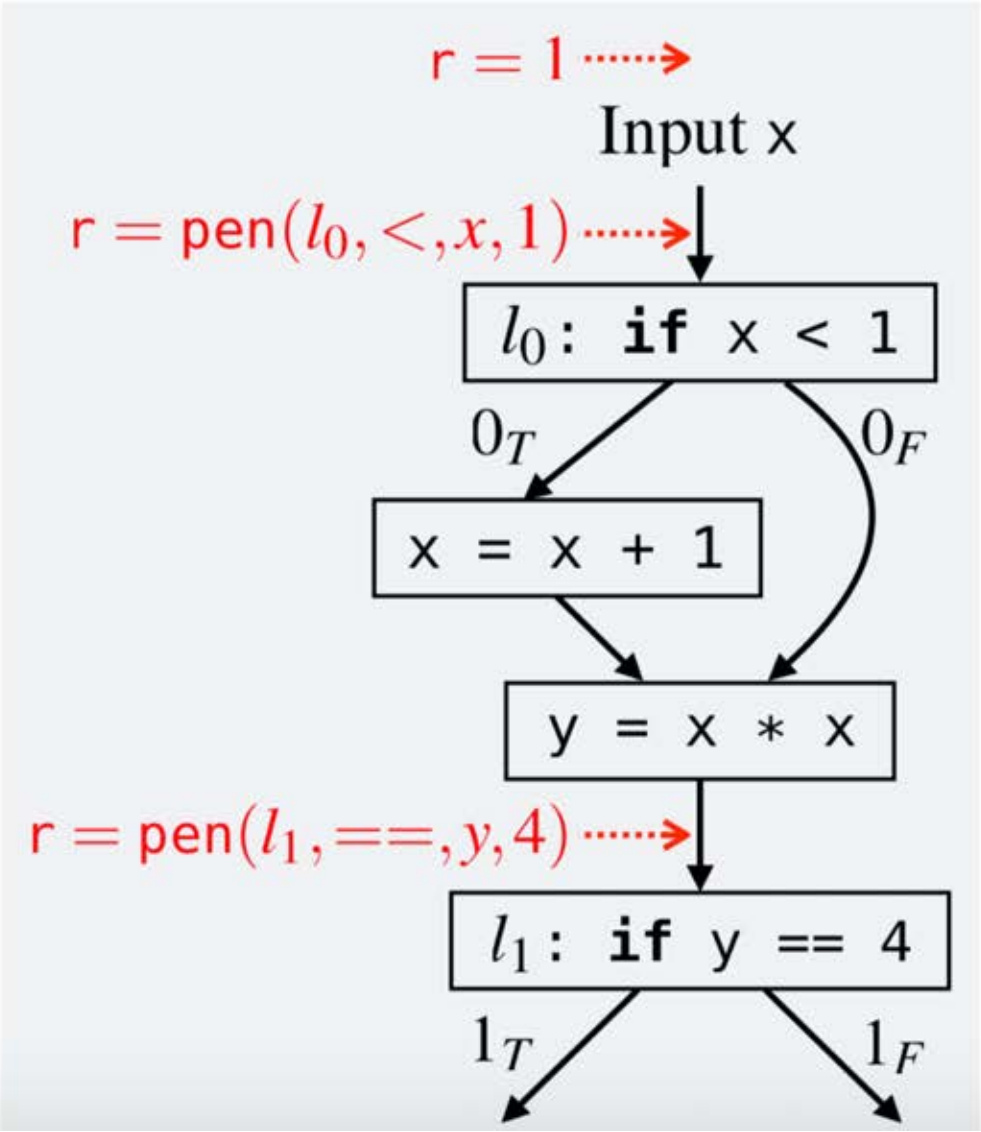
```
void FOO (double x){
l0: if (x < 1)
        x++;
    double y = x * x;
l1: if (y == 4)
        ...
}
```



Step 1: Construct FOO_R

covered at l_i	$pen(l_i, op, a, b)$
$\emptyset$	0
$\{i_F\}$	$R_{a\ op\ b}$
$\{i_T\}$	$R_{\neg(a\ op\ b)}$
$\{i_T, i_F\}$	r

- r: global variable
- FOO_R : $x \rightarrow$ r
- $R_{a\ op\ b}$:Branch distance



Branch distance $R_{a\ op\ b}$

A helper function to quantify how far a and b are from attaining branch *a op b*.

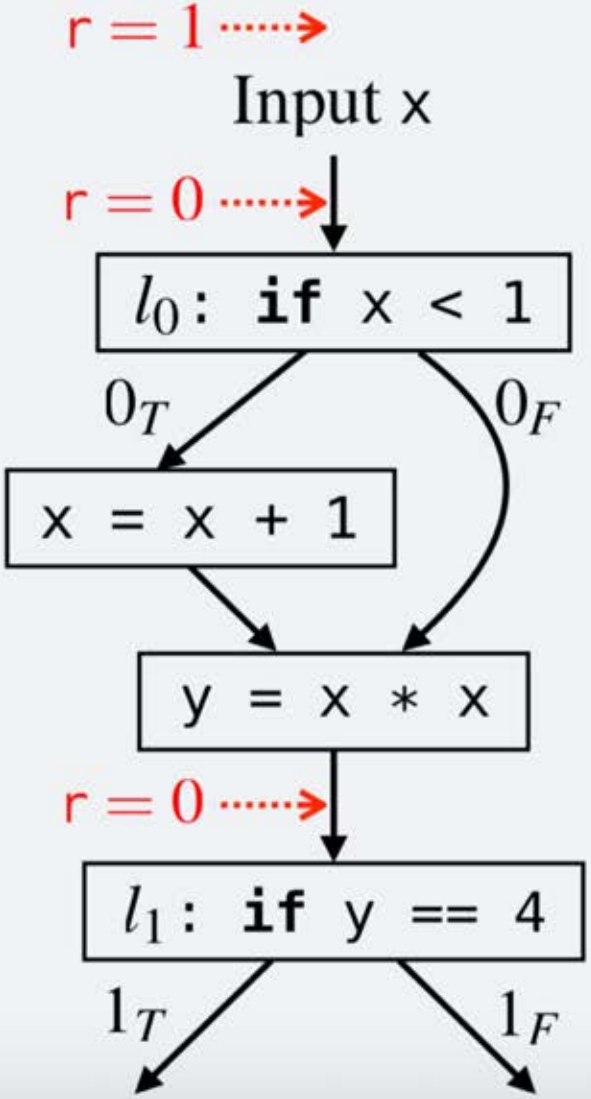
$R_{a==b}$ defined as $(a-b)^2$

$R_{a\geq b}$ defined as $(a \geq b)\ ?\ 0 : (a-b)^2$

Step 2: Minimize FOO_R

covered at l_i	$pen(l_i, op, a, b)$
$\emptyset$	0
$\{i_F\}$	$R_{a\ op\ b}$
$\{i_T\}$	$R_{\neg(a\ op\ b)}$
$\{i_T, i_F\}$	r

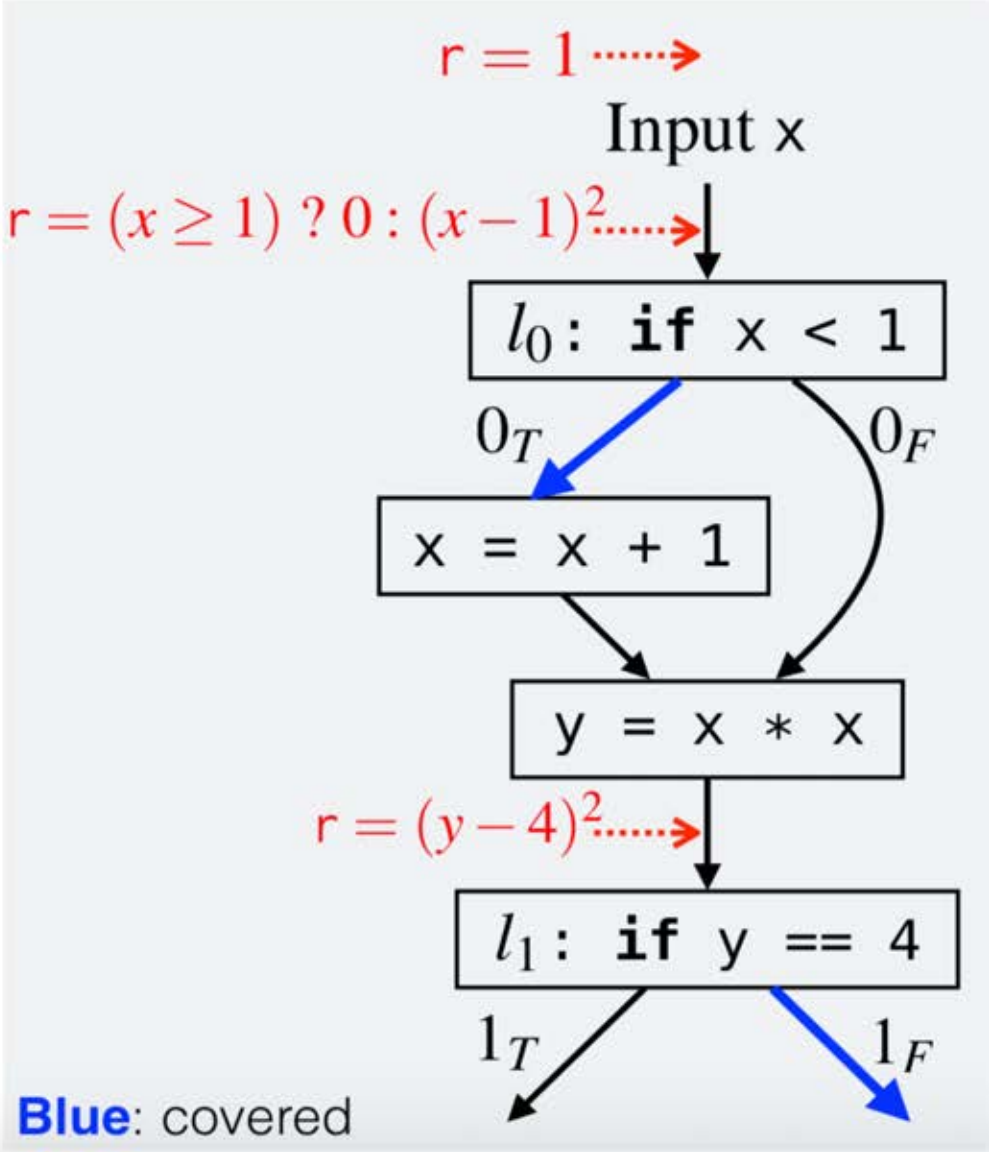
- No branch is covered
- Any input is a minimum point
- Assume $x^* = 0.7$



Step 2: Minimize FOO_R

covered at l_i	$pen(l_i, op, a, b)$
$\emptyset$	0
$\{i_F\}$	$R_{a\ op\ b}$
$\{i_T\}$	$R_{\neg(a\ op\ b)}$
$\{i_T, i_F\}$	r

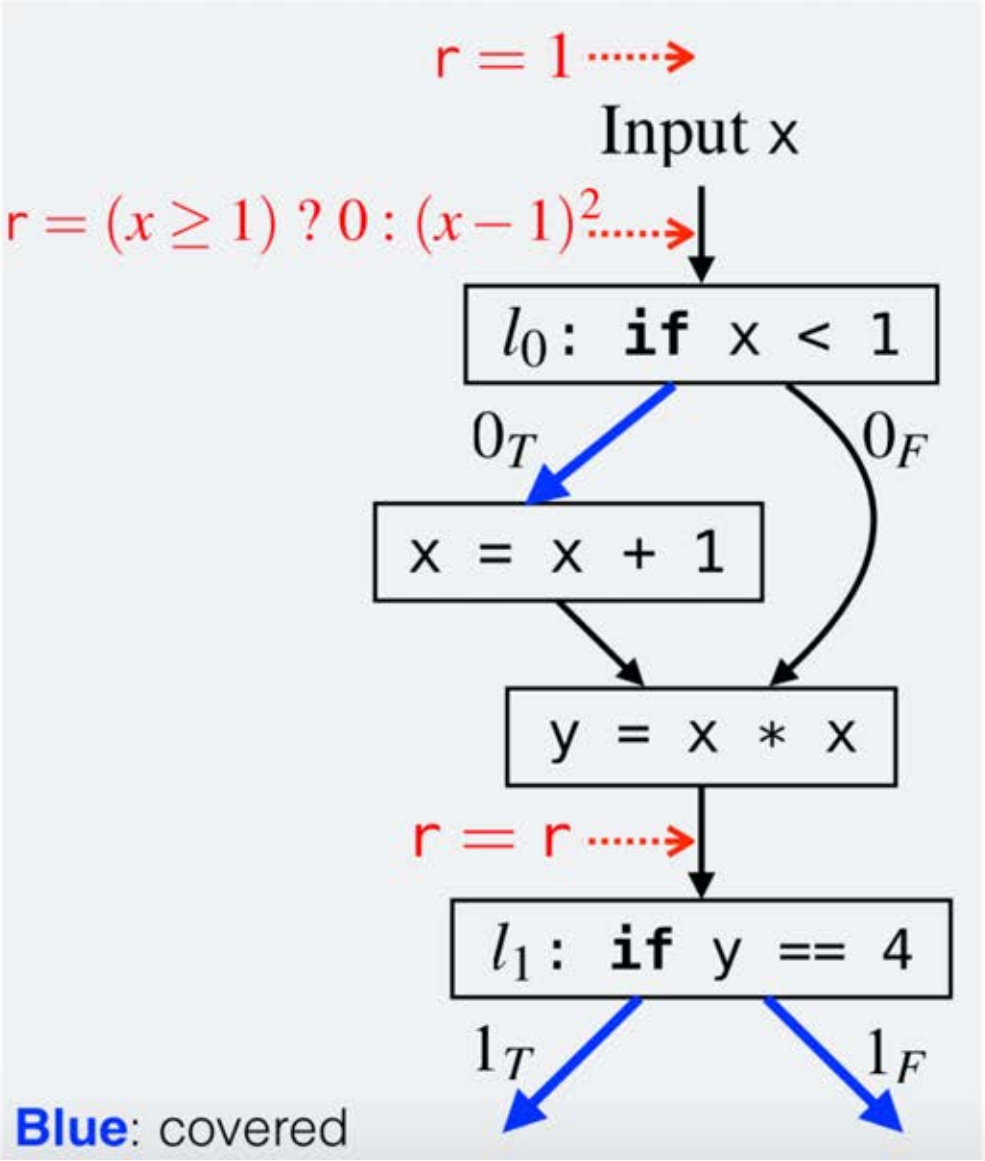
- $1_F, 0_T$ are covered
- FOO_R attains minimum at -3 or 2
- Assume $x^* = -3$



Step 2: Minimize FOO_R

covered at l_i	$pen(l_i, op, a, b)$
$\emptyset$	0
$\{i_F\}$	$R_{a\ op\ b}$
$\{i_T\}$	$R_{\neg(a\ op\ b)}$
$\{i_T, i_F\}$	r

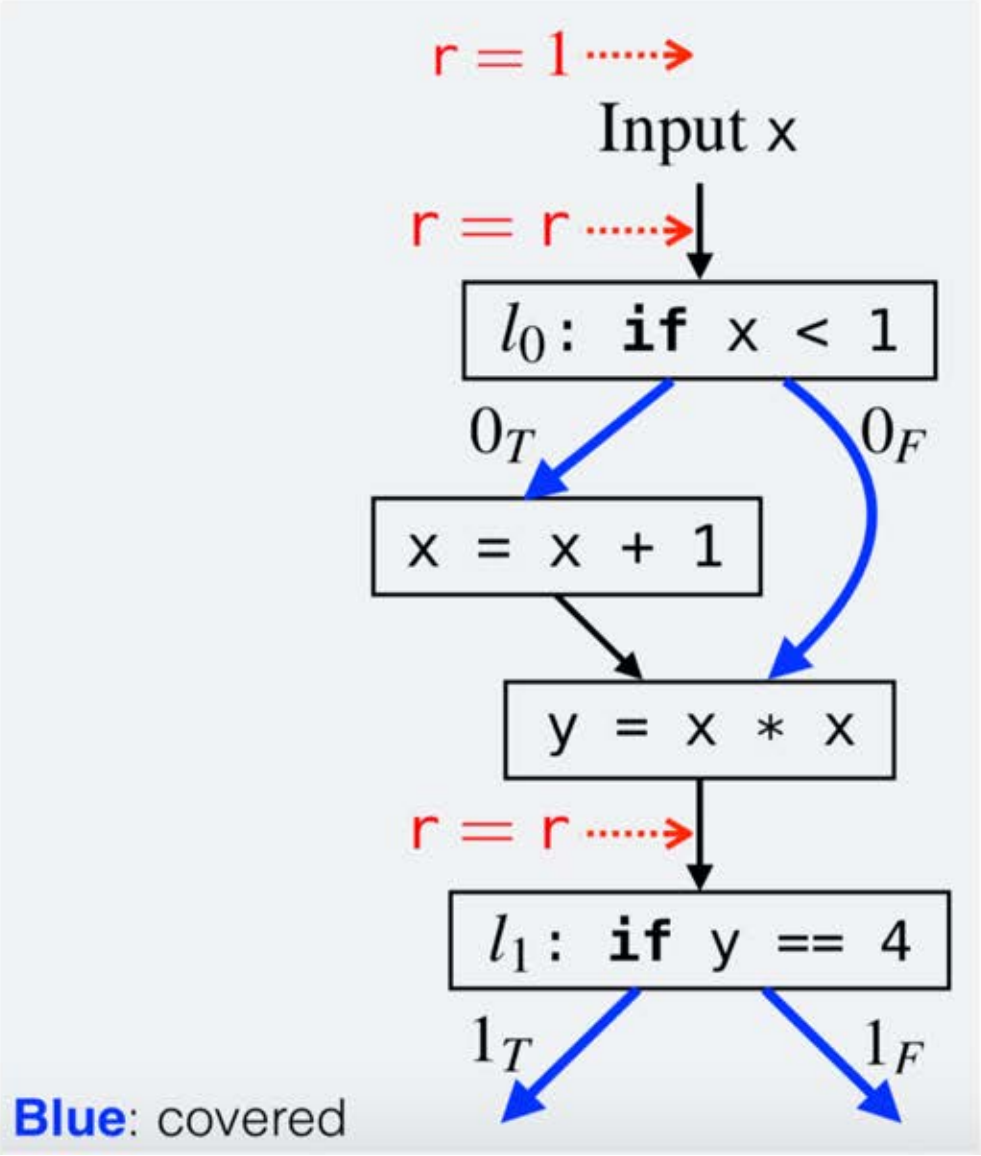
- $1_F, 1_T, 0_T$ are covered
- FOO_R attains minimum at ≥ 1
- Assume $x^* = 5.1$



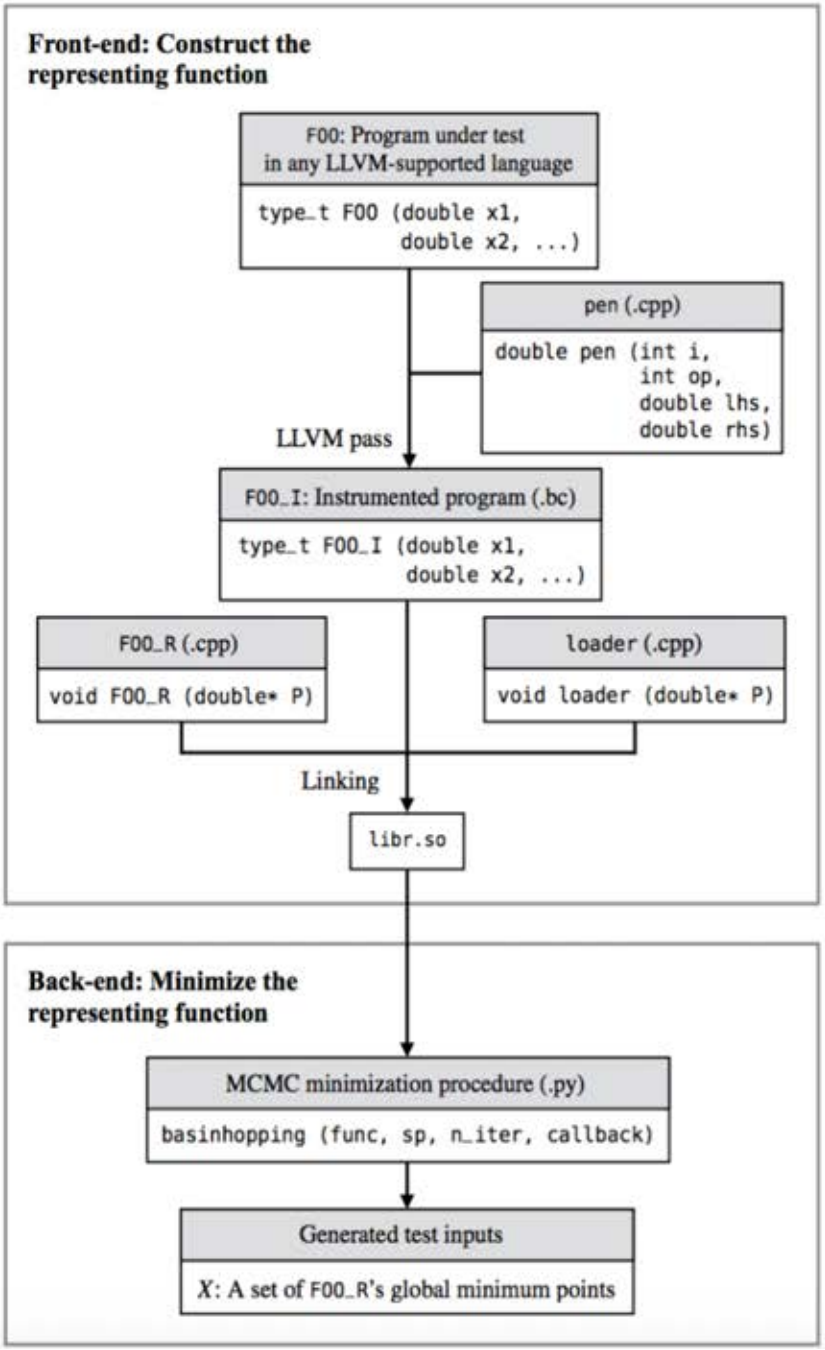
Step 2: Minimize FOO_R

covered at l_i	$pen(l_i, op, a, b)$
$\emptyset$	0
$\{i_F\}$	$R_{a\ op\ b}$
$\{i_T\}$	$R_{\neg(a\ op\ b)}$
$\{i_T, i_F\}$	r

- All branches are covered
- $\forall x, \mathsf{FOO\_R}(x) = 1$
- Termination



Our implementation CoverMe



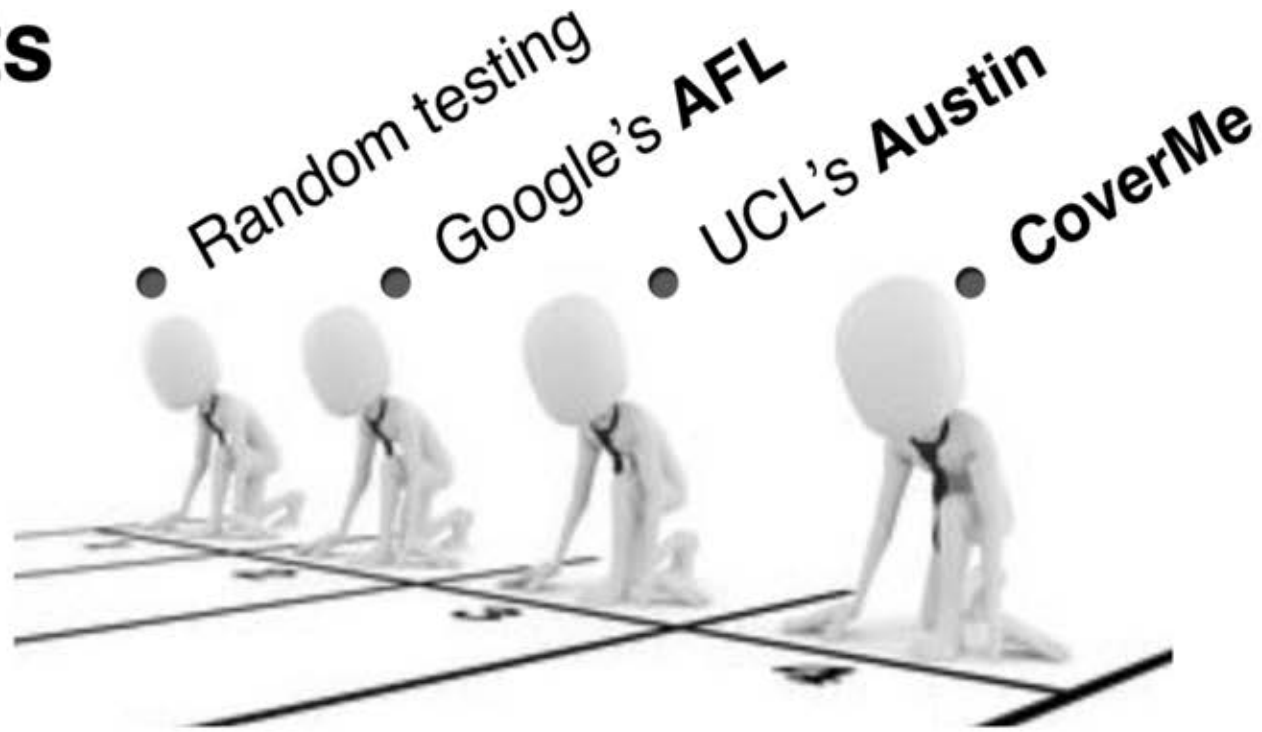
- **Front-end**: Program transformation in LLVM
- **Back-end**: Basinhopping

Experiments

Benchmarks: **Fdlibm**

- Sun's math library
- Reference for Java SE 8's math library
- Used in Matlab, JavaScript and Android
- Heavy on branches (max=114, avg=23)

Results



CoverMe covers

- ≈ 90% branches in 7 seconds
- ≈ 18% more branches than AFL with 1/10 time
- ≈ 40% more branches than Austin with

speedups of several orders of magnitudes

ME in the long run

- Offers a new general analysis paradigm
- Complements existing approaches
  - Random concrete execution (CE)
  - Symbolic execution (SE)
  - Abstract execution (AE)

Peeking into the future …

What is the **key mission** of Computer Science?

To help people **turn** **creative**

ideas **into** **working systems**

Software research is

central to this mission


NATO Software Engineering
Conference 1968

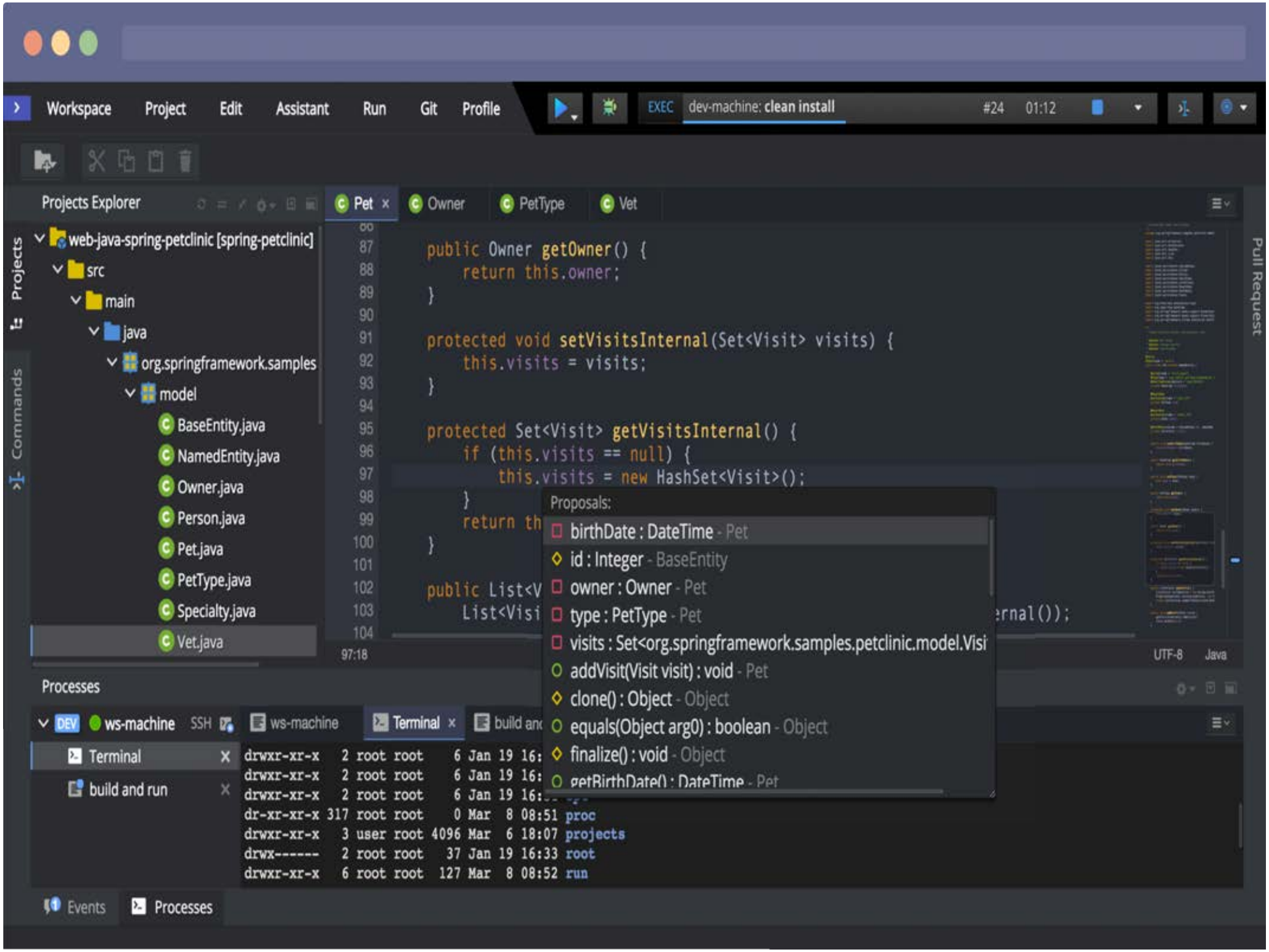
A lot of progress

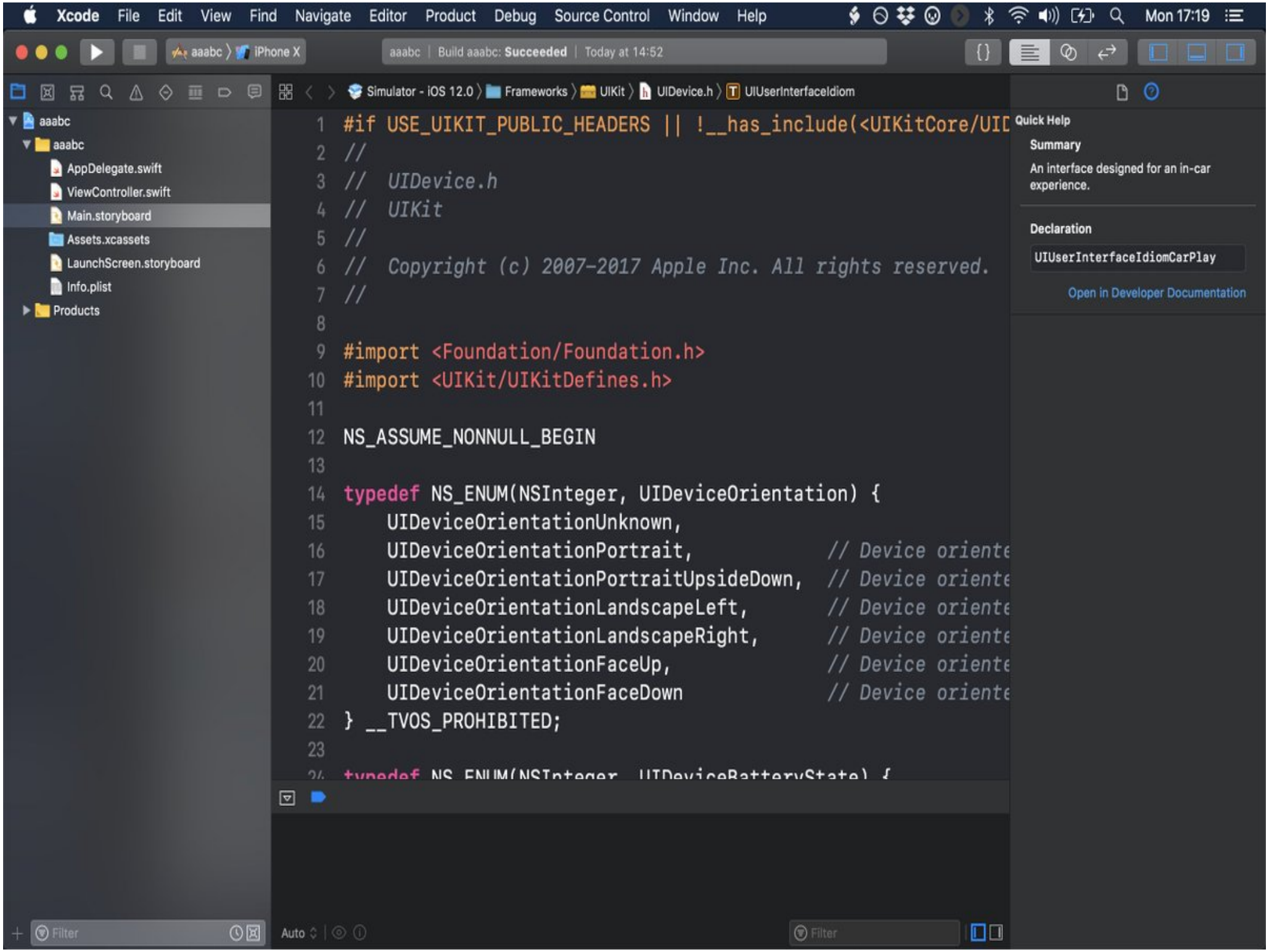
to celebrate for!

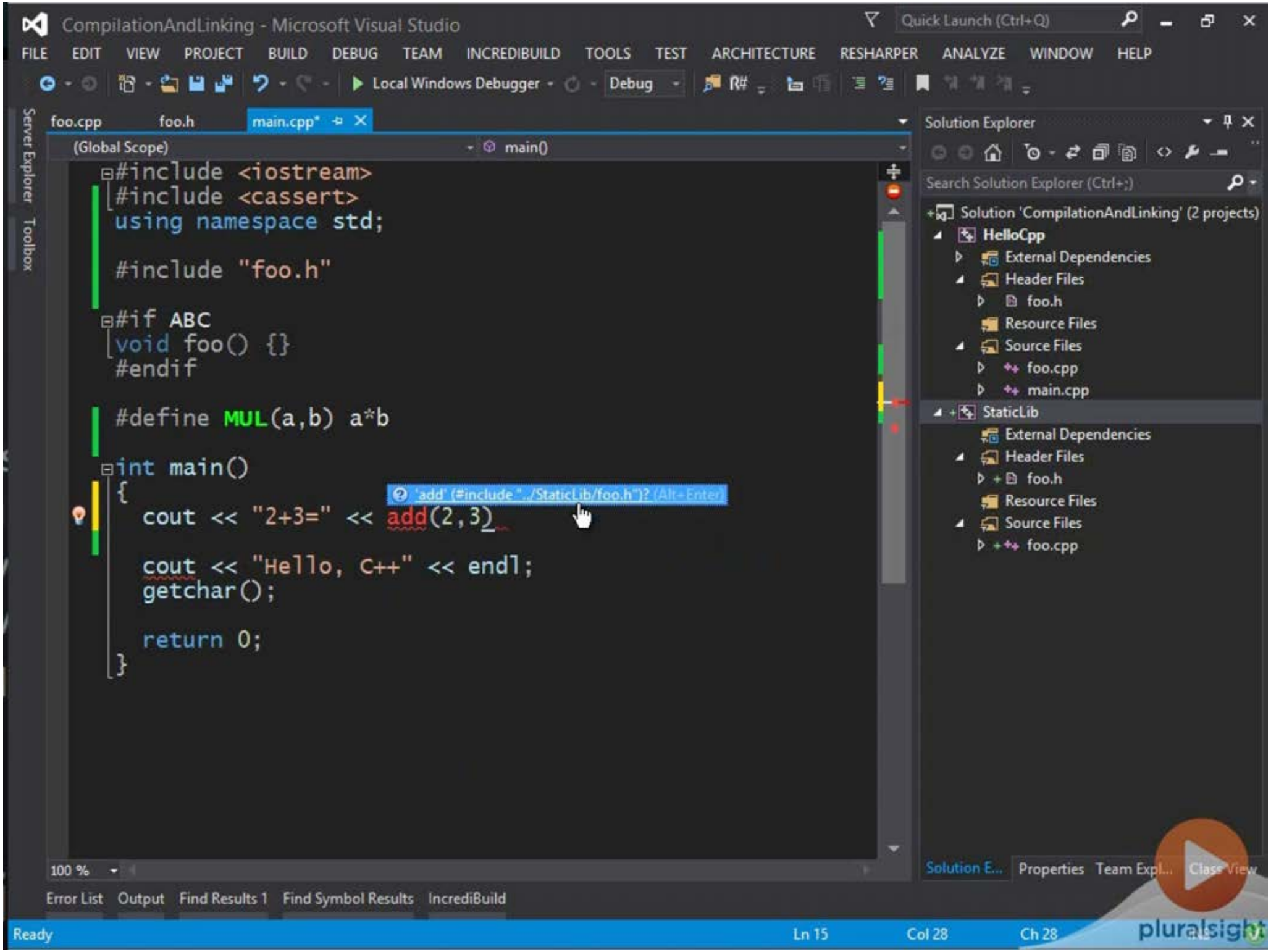
But …

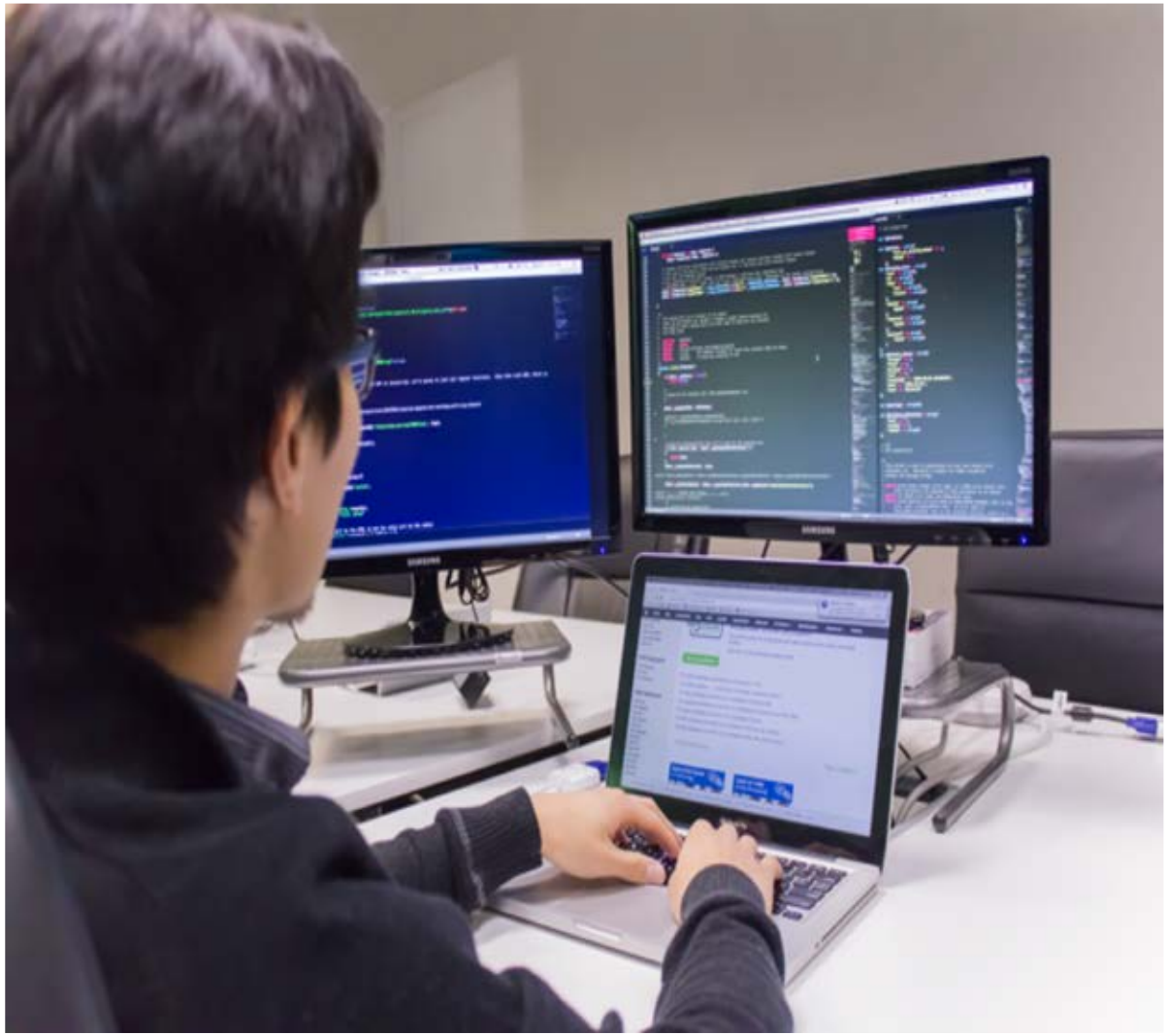
```c
#include <stdio.h>
#include <string.h>

int main(){
    char *word = "everest";
    char reverseword[strlen(word)+1];
    unsigned int letters_remaining = strlen(word);
    char *wordpointer = &word[strlen(word)-1];
    int i = 0;
    while(letters_remaining > 0){
        reverseword[i++] = *wordpointer--;
        letters_remaining--;
    }
    reverseword[strlen(word)] = '\0';
    printf("So the reversed word is %s\n",reverseword);
    return 0;
}
```

```
1  #ifndef __ASM_ALPHA_FPU_H
2  #define __ASM_ALPHA_FPU_H
3
4  #include <asm/special_insns.h>
5  #include <uapi/asm/fpu.h>
6
7  /* The following two functions don't need trapb/excb instructions
8     around the mf_fpcr/mt_fpcr instructions because (a) the kernel
9     never generates arithmetic faults and (b) call_pal instructions
10    are implied trap barriers.  */
11
12 static inline unsigned long
13 rdfpcr(void)
14 {
15     unsigned long tmp, ret;
16
17 #if defined(CONFIG_ALPHA_EV6) || defined(CONFIG_ALPHA_EV67)
18     __asm__ __volatile__ (
19         "ftoit $f0,%0\n\t"
20         "mf_fpcr $f0\n\t"
21         "ftoit $f0,%1\n\t"
22         "itoft %0,$f0"
23         : "=r"(tmp), "=r"(ret));
24 #else
25     __asm__ __volatile__ (
26         "stt $f0,%0\n\t"
27         "mf_fpcr $f0\n\t"
28         "stt $f0,%1\n\t"
29         "ldt $f0,%0"
30         : "=m"(tmp), "=m"(ret));
31 #endif
32
33     return ret;
34 }
35
36 static inline void
37 wrfpcr(unsigned long val)
38 {
39     unsigned long tmp;
40
41 #if defined(CONFIG_ALPHA_EV6) || defined(CONFIG_ALPHA_EV67)
42     __asm__ __volatile__ (
43         "ftoit $f0,%0\n\t"
44         "itoft %1,$f0\n\t"
```

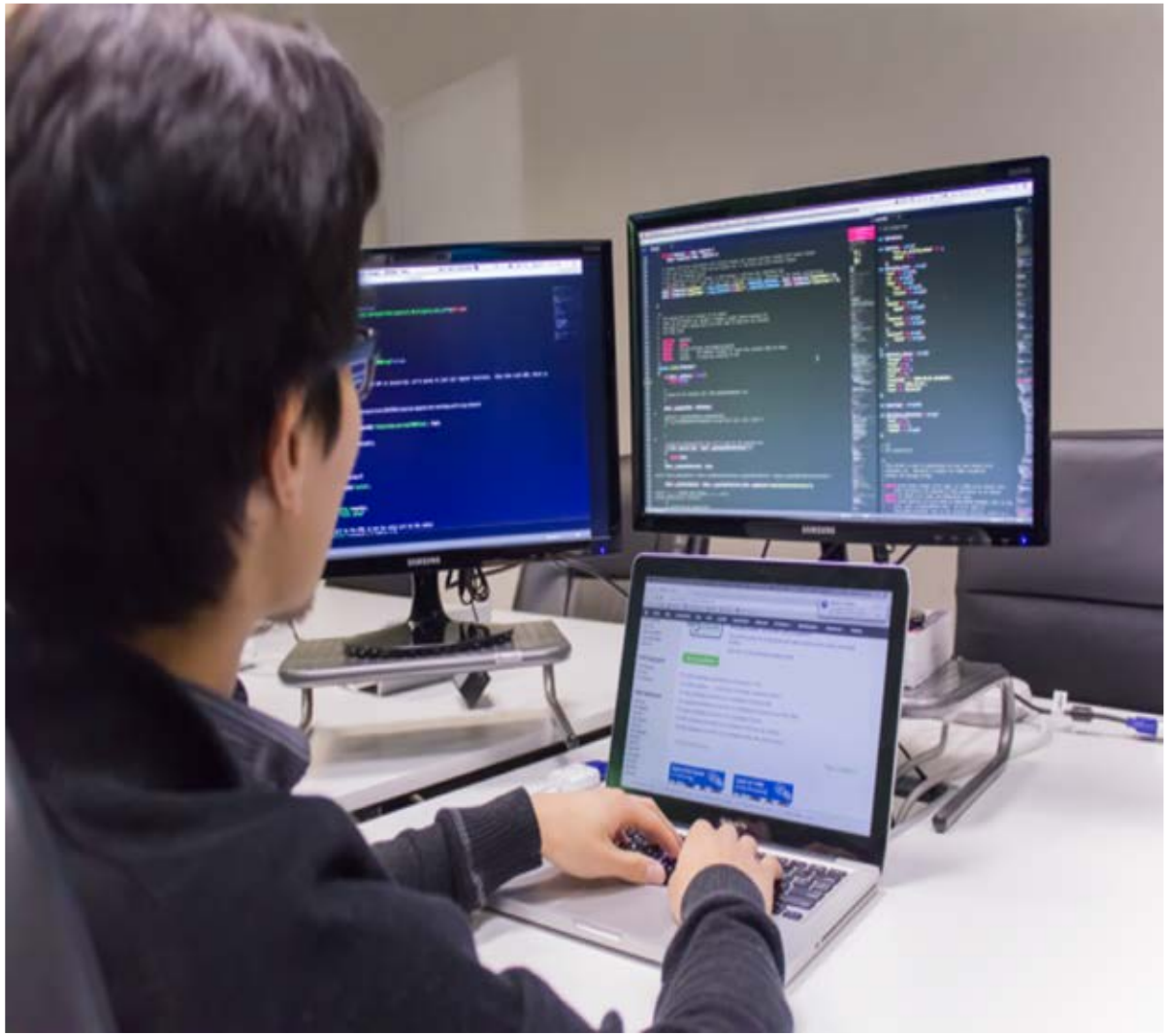

Workspace
Project
Edit
Assistant
Run
Git
Profile
EXEC
dev-machine: clean install
#24
01:12
Projects Explorer
web-java-spring-petclinic [spring-petclinic]
src
main
java
org.springframework.samples
model
BaseEntity.java
NamedEntity.java
Owner.java
Person.java
Pet.java
PetType.java
Specialty.java
Vet.java
Pet
Owner
PetType
Vet
public Owner getOwner() {
return this.owner;
}
protected void setVisitsInternal(Set<Visit> visits) {
this.visits = visits;
}
protected Set<Visit> getVisitsInternal() {
if (this.visits == null) {
this.visits = new HashSet<Visit>();
}
return th
}
public List<V
List<Visi
ernal());
Proposals:
birthDate : DateTime - Pet
id : Integer - BaseEntity
owner : Owner - Pet
type : PetType - Pet
visits : Set<org.springframework.samples.petclinic.model.Visi
addVisit(Visit visit) : void - Pet
clone() : Object - Object
equals(Object arg0) : boolean - Object
finalize() : void - Object
getBirthDate() : DateTime - Pet
97:18
UTF-8
Java
Processes
DEV
ws-machine
SSH
ws-machine
Terminal
build and run
drwxr-xr-x 2 root root 6 Jan 19 16:
drwxr-xr-x 2 root root 6 Jan 19 16:
drwxr-xr-x 2 root root 6 Jan 19 16:
dr-xr-xr-x 317 root root 0 Mar 8 08:51 proc
drwxr-xr-x 3 user root 4096 Mar 6 18:07 projects
drwx------ 2 root root 37 Jan 19 16:33 root
drwxr-xr-x 6 root root 127 Mar 8 08:52 run
Events
Processes
Projects
Commands
Pull Request

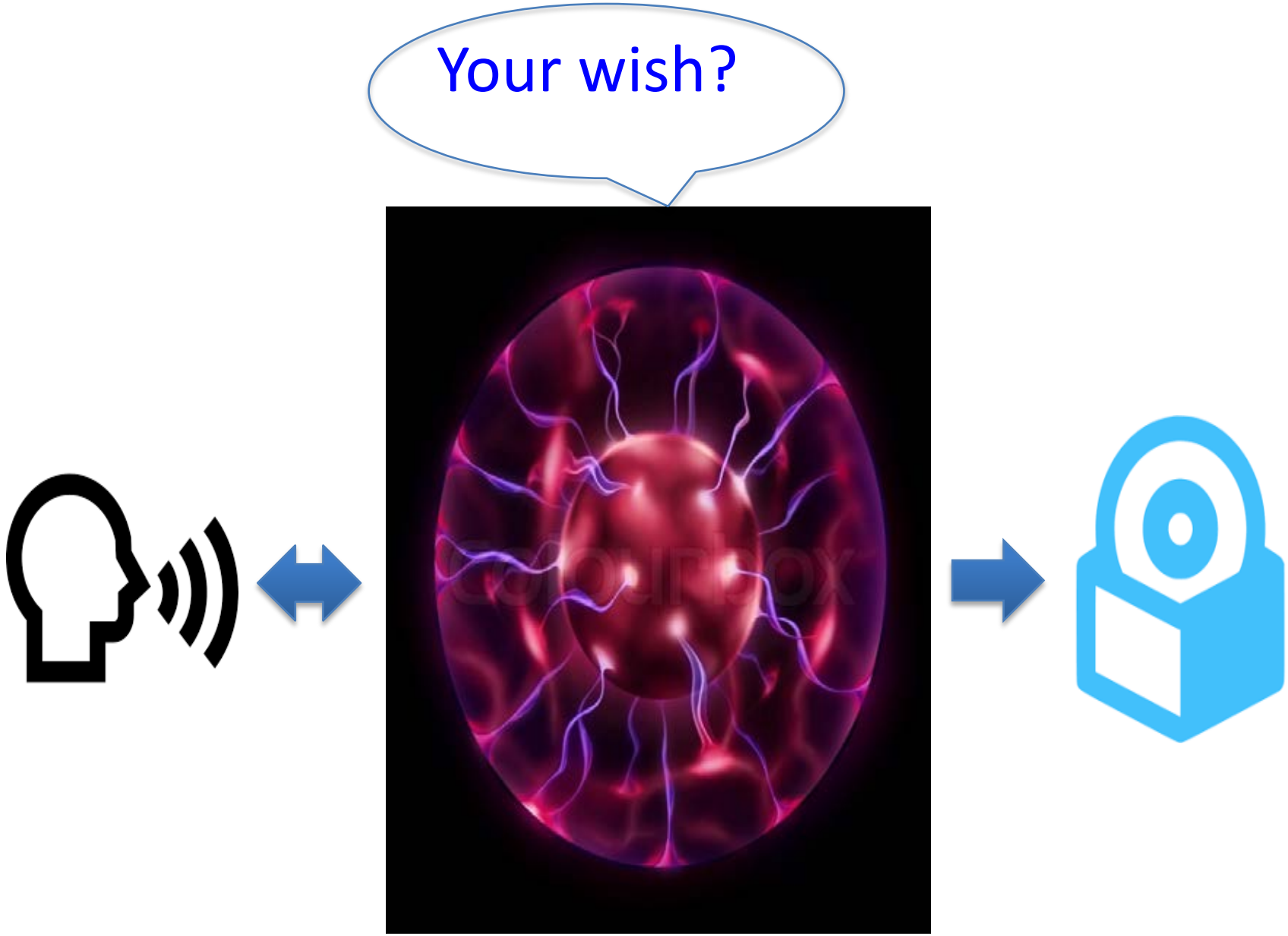

Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help
Mon 17:19
aaabc 〉 iPhone X
aaabc | Build aaabc: Succeeded | Today at 14:52
Simulator - iOS 12.0 〉 Frameworks 〉 UIKit 〉 UIDevice.h 〉 UIUserInterfaceIdiom
aaabc
aaabc
AppDelegate.swift
ViewController.swift
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
Products
#if USE_UIKIT_PUBLIC_HEADERS || !__has_include(<UIKitCore/UIDevice.h>)
//
// UIDevice.h
// UIKit
//
// Copyright (c) 2007-2017 Apple Inc. All rights reserved.
//
#import <Foundation/Foundation.h>
#import <UIKit/UIKitDefines.h>
NS_ASSUME_NONNULL_BEGIN
typedef NS_ENUM(NSInteger, UIDeviceOrientation) {
UIDeviceOrientationUnknown,
UIDeviceOrientationPortrait, // Device oriente
UIDeviceOrientationPortraitUpsideDown, // Device oriente
UIDeviceOrientationLandscapeLeft, // Device oriente
UIDeviceOrientationLandscapeRight, // Device oriente
UIDeviceOrientationFaceUp, // Device oriente
UIDeviceOrientationFaceDown // Device oriente
} __TVOS_PROHIBITED;
Quick Help
Summary
An interface designed for an in-car experience.
Declaration
UIUserInterfaceIdiomCarPlay
Open in Developer Documentation
Filter
Auto
Filter

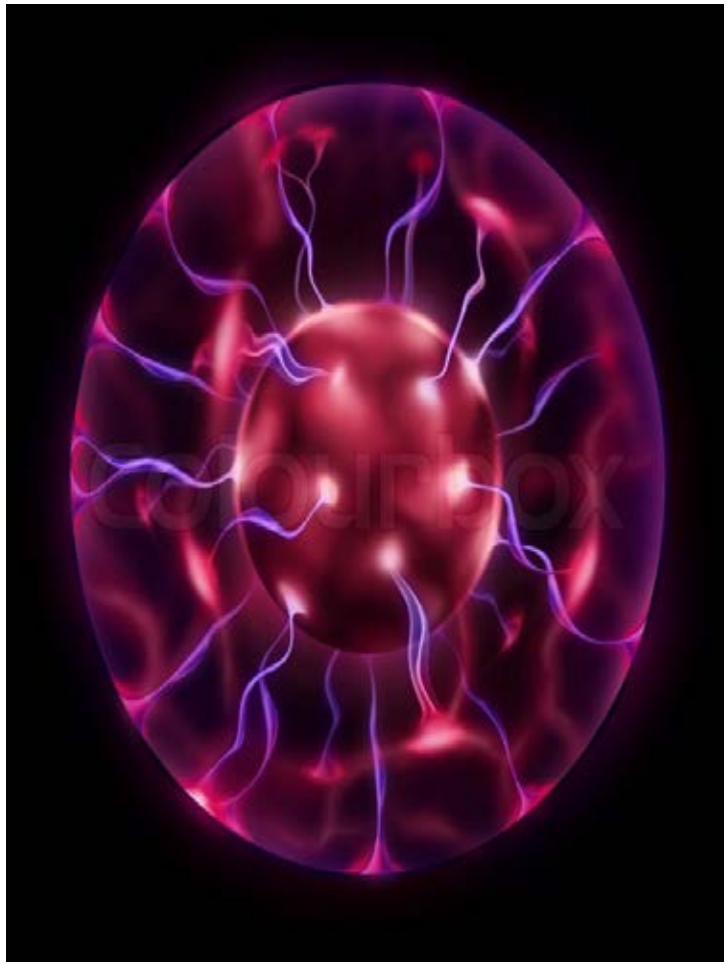

CompilationAndLinking - Microsoft Visual Studio
Quick Launch (Ctrl+Q)
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM INCREDIBUILD TOOLS TEST ARCHITECTURE RESHARPER ANALYZE WINDOW HELP
Local Windows Debugger
Debug
foo.cpp
foo.h
main.cpp*
(Global Scope)
main()
#include <iostream>
#include <cassert>
using namespace std;
#include "foo.h"
#if ABC
void foo() {}
#endif
#define MUL(a,b) a*b
int main()
{
cout << "2+3=" << add(2,3)
'add' (#include "../StaticLib/foo.h")? (Alt+Enter)
cout << "Hello, C++" << endl;
getchar();
return 0;
}
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'CompilationAndLinking' (2 projects)
HelloCpp
External Dependencies
Header Files
foo.h
Resource Files
Source Files
foo.cpp
main.cpp
StaticLib
External Dependencies
Header Files
foo.h
Resource Files
Source Files
foo.cpp
Server Explorer
Toolbox
100 %
Error List
Output
Find Results 1
Find Symbol Results
IncrediBuild
Ready
Ln 15
Col 28
Ch 28
pluralsight



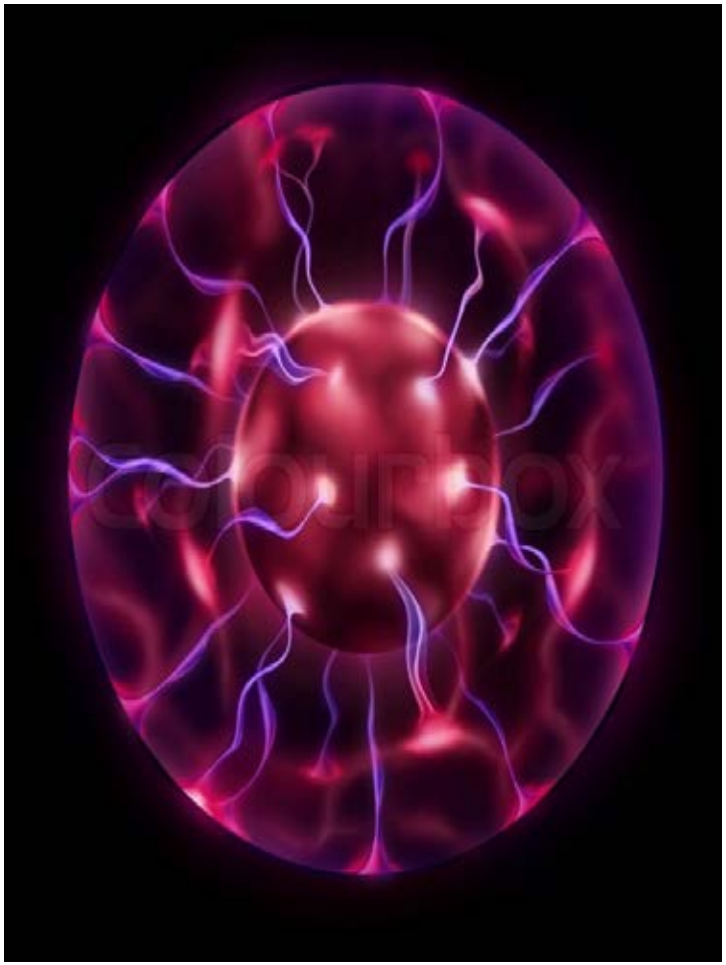
Can we move beyond “coding”?




Your wish?



Hardest: communicate the wish

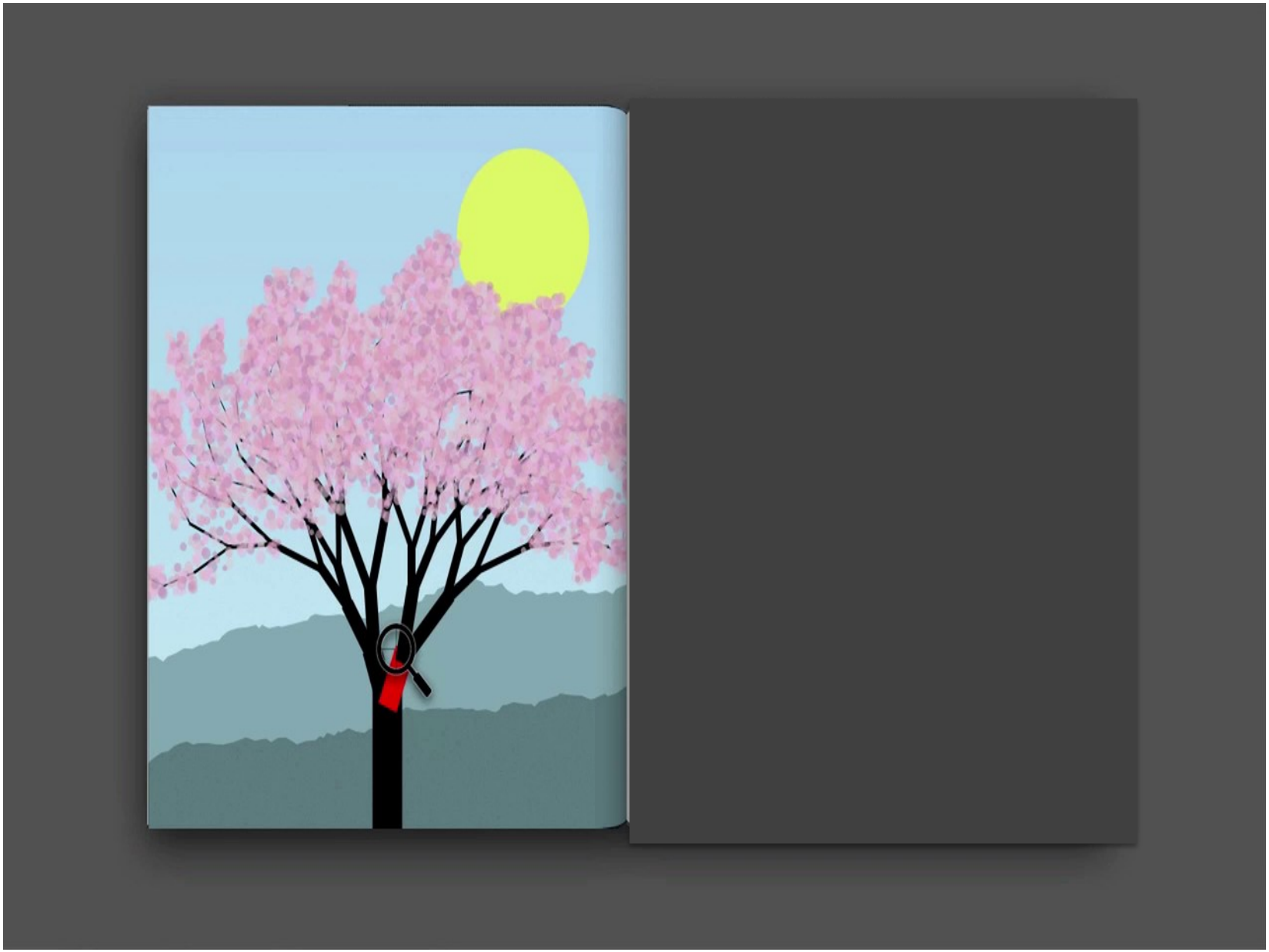


```
function drawTree () {
    var blossomPoints = [];

    resetRandom();
    drawBranches(0, -Math.PI/2, canvasWidth/2, canvasHeight, 30,

    resetRandom();
    drawBlossoms(blossomPoints);
}

function drawBranches (i,angle,x,y,width,blossomPoints) {
    ctx.save();

    var length = tween(i, 1, 62, 12, 3) * random(0.7, 1.3);
    if (i == 0) { length = 107; }

    ctx.translate(x,y);
    ctx.rotate(angle);
    ctx.fillStyle = "#000";
    ctx.fillRect(0, -width/2, length, width);

    ctx.restore();

    var tipX = x + (length - width/2) * Math.cos(angle);
    var tipY = y + (length - width/2) * Math.sin(angle);

    if (i > 4) {
        blossomPoints.push([x,y,tipX,tipY]);
    }

    if (i < 6) {
        drawBranches(i + 1, angle + random(-0.15, -0.05) * Math
        drawBranches(i + 1, angle + random( 0.15,  0.05) * Math
    }
    else if (i < 12) {
        drawBranches(i + 1, angle + random( 0.25, -0.05) * Math.
    }
}
```

```
    var length = tween(i, 1, 62, 12, 3) + random(0.7, 1.3);
    if (i == 0) { length = 107; }

    ctx.translate(x,y);
    ctx.rotate(angle);
    ctx.fillStyle = color;
    ctx.fillRect(153, -width/2, length, width);

    ctx.restore();

    var tipX = x + (length - width/2) * Math.cos(angle);
    var tipY = y + (length - width/2) * Math.sin(angle);

    if (i > 4) {
        blossomPoints.push([x,y,tipX,tipY]);
    }

    if (i < 6) {
        drawBranches(i + 1, angle + random(-0.15, -0.05) * Math.
        drawBranches(i + 1, angle + random( 0.15,  0.05) * Math.
    }
    else if (i < 12) {
        drawBranches(i + 1, angle + random( 0.25, -0.05) * Math.
    }
}

function drawBlossoms (blossomPoints) {
    var colors = ["#f5ceea", "#e8d9e4", "#f7c9f3", "#ebb4cc", "#
    ctx.globalAlpha = 0.60;

    for (var i = 0; i < blossomPoints.length; i++) {
        var p = blossomPoints[i];
        for (var j = 0; j < 16; j++) {
            var x = lerp(p[0], p[2], random(0,1)) + random(-10,1
            var y = lerp(p[1], p[3], random(0,1)) + random(-10,1

            ctx.fillStyle = colors[Math.floor(random(0,colors.le
            ctx.fillCircle(x, y, random(2,5));
        }
```

Bret Victor: Inventing on Principle

```
function drawTree () {
    var blossomPoints = [];

    resetRandom();
    drawBranches(0, -Math.PI/2, canvasWidth/2, canvasHeight, 30,

    resetRandom();
    drawBlossoms(blossomPoints);
}

function drawBranches (i,angle,x,y,width,blossomPoints) {
    ctx.save();

    var length = tween(i, 1, 62, 12, 3) * random(0.7, 1.3);
    if (i == 0) { length = 107; }

    ctx.translate(x,y);
    ctx.rotate(angle);
    ctx.fillStyle = "#000";
    ctx.fillRect(0, -width/2, length, width);

    ctx.restore();

    var tipX = x + (length - width/2) * Math.cos(angle);
    var tipY = y + (length - width/2) * Math.sin(angle);

    if (i > 4) {
        blossomPoints.push([x,y,tipX,tipY]);
    }

    if (i < 6) {
        drawBranches(i + 1, angle + random(-0.15, -0.05) * Math
        drawBranches(i + 1, angle + random( 0.15,  0.05) * Math
    }
    else if (i < 12) {
        drawBranches(i + 1, angle + random( 0.25, -0.05) * Math.
    }
}
```



Goal is the object, not the code

Can we directly
manipulate & explore the object
to express the “wish”?

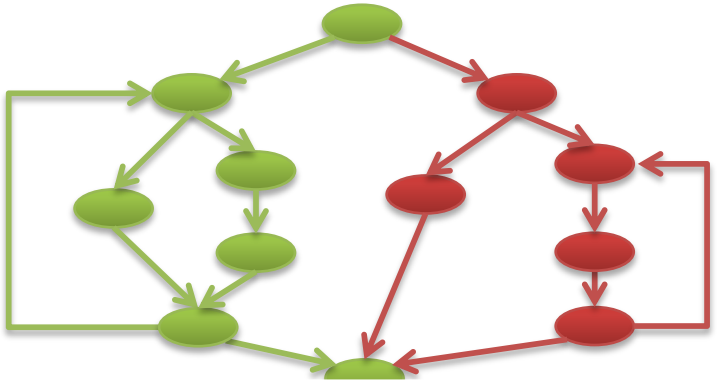
Can we directly
manipulate & explore the object
to express the “wish”?

Perhaps via visualization & virtual reality?

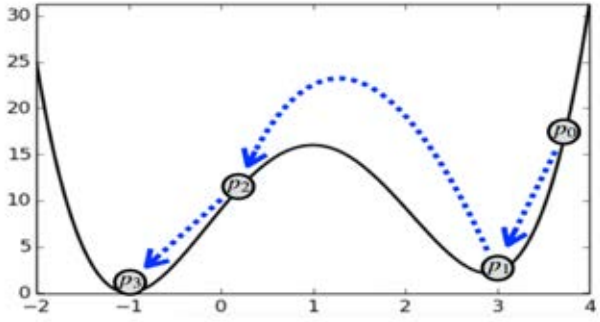

PERSPECTIVE
IS
EVERYTHING


1/2
1/2

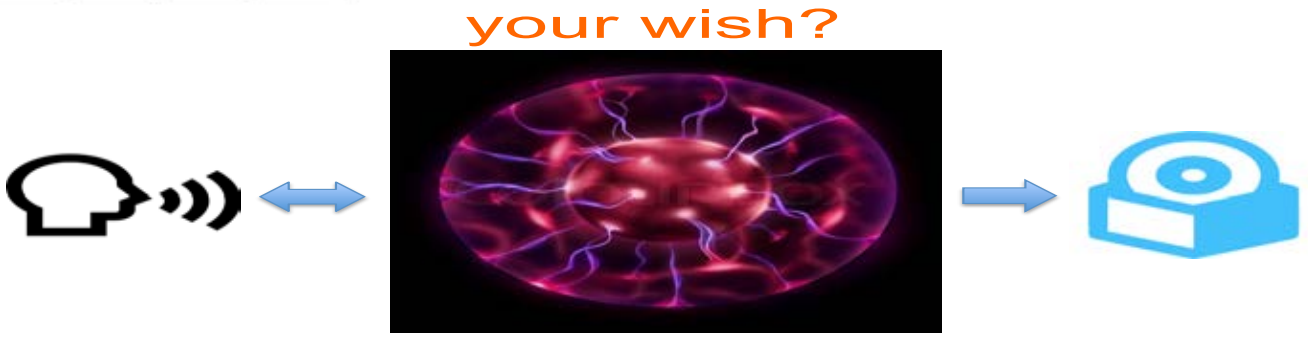
Advancing Software Analysis via Changed Perspectives



	GCC	LLVM	TOTAL
Reported	841	781	**1622**
Fixed	612	419	**1031**



Benchmark			Satisfiability				Time (seconds)			
SMT2-LIB program	size (byte)	#var	MathSat	Z3	Coral	XSat	MathSat	Z3	Coral	XSat
SUMMARY			-	100.0%	54.6%	100.0%	2014.75	2290.05	1.38	2.80



Thank you!