

FINDING DEEP COMPILER BUGS

VIA

GUIDED STOCHASTIC
PROGRAM MUTATION

VU LE MICROSOFT*

CHENGNIAN SUN UC DAVIS

ZHENDONG SU UC DAVIS

(*: WORK DONE @ UC DAVIS)

WHY COMPILER BUGS?

affect **ALL** software



GCC BUG 61383

Miscompilation Bug

```
int a, c, d, e = 1, f;
int fn1 () {
    int h;
    for (; d < 1; d = e) {
        h = (f == 0) ? 0 : 1 % f;
        if (f < 1) c = 0;
        else if (h) break;
    }
}
int main () {
    fn1 ();
    return 0;
}
```

>_

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

```
$ gcc -O2 test.c ; ./a.out
```

Floating point exception (core
dumped)

https://gcc.gnu.org/bugzilla/show_bug.cgi?id=61383



LLVM BUG 18615

Crashing Bug

```
int a;  
struct S0 {  
    int f0; int f1; int f2;  
};  
void fn1 () {  
    int b = -1;  
    struct S0 f[1];  
    if (a) {  
        f[0] = f[b];  
    }  
}  
int main () {  
    fn1 ();  
    return 0;  
}
```

>_

```
$ clang -O0 test.c
```

```
$ clang -O1 test.c
```

```
clang: Assertion failed.
```

```
clang: error: Aborted (core dumped)
```

https://llvm.org/bugs/show_bug.cgi?id=18615



COMPILER TESTING CHALLENGES

- Generate **valid** test programs
 - No undefined behavior
- Determine the **semantics** of test programs
 - No referencing compilers

EQUIVALENCE MODULO INPUTS*

(EMI)

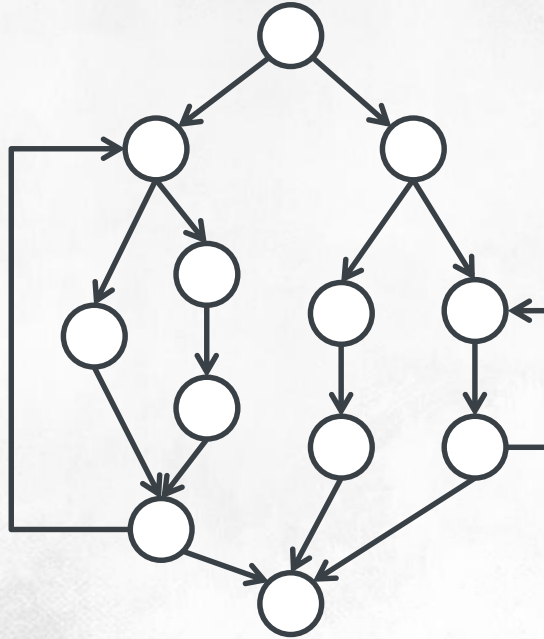
generates valid, “equivalent” programs
from existing programs

*: V. Le, M. Afshari, and Z. Su. *Compiler validation via equivalence modulo inputs*. PLDI '14



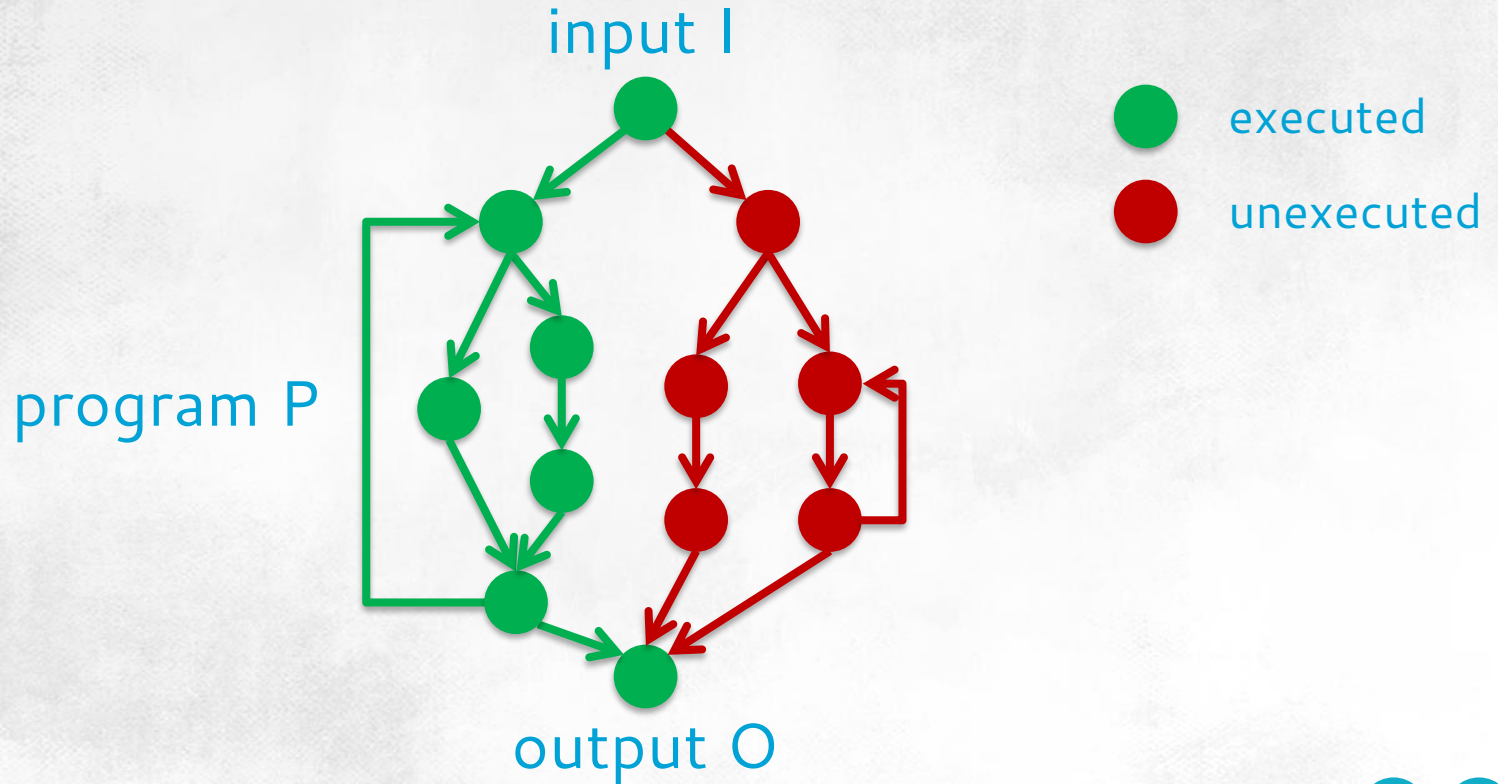
ILLUSTRATION

input I

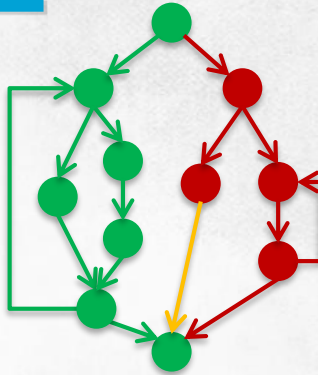
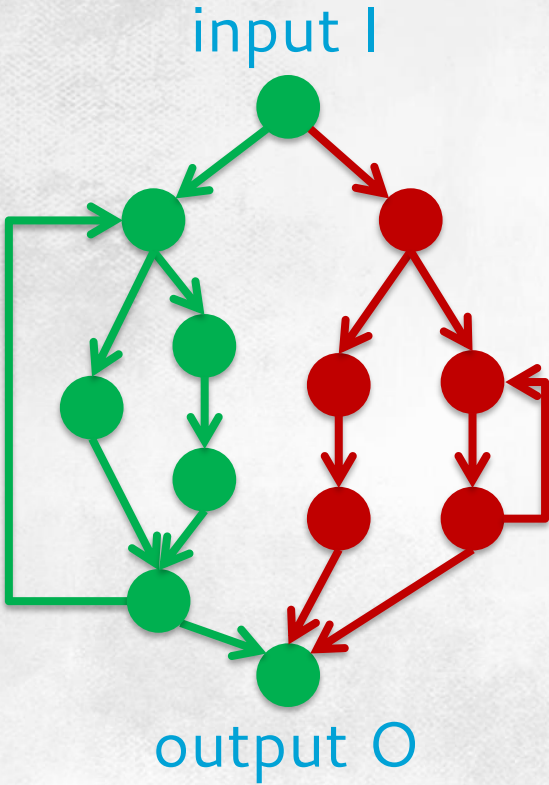


program P

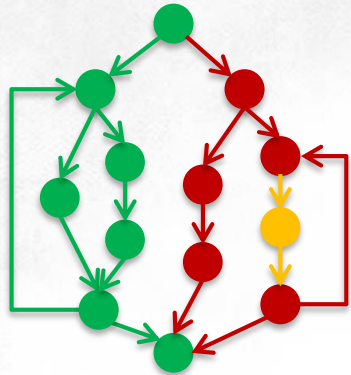
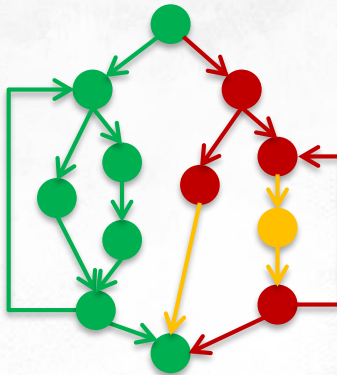
PROFILE



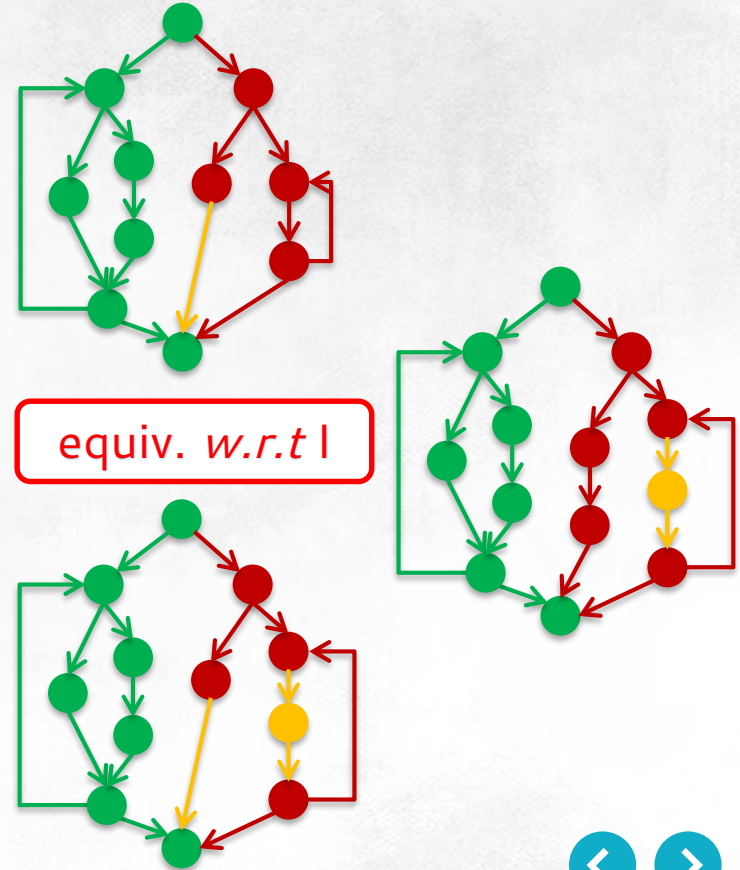
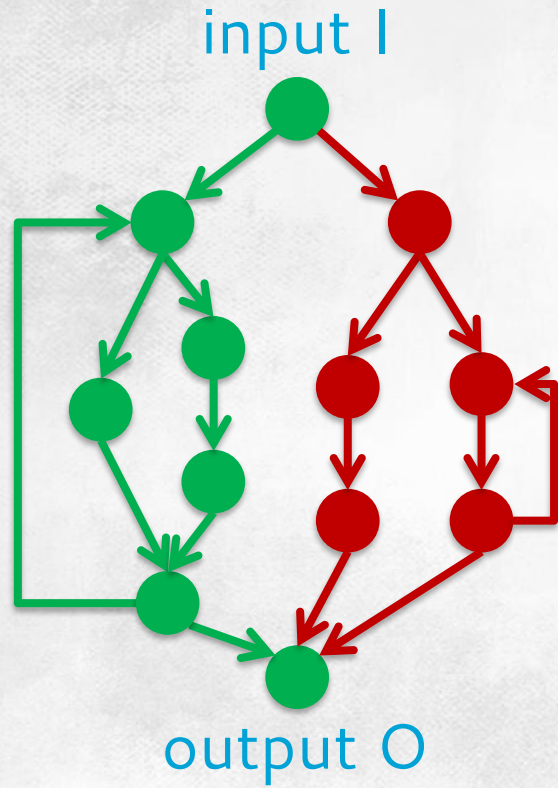
MUTATE



□ □ □ □ □



EMI



ORION*

Naïve EMI Instantiation

- Randomly **removes** unexecuted code

✗ Limitation

- Limited number of variants
- Limited control- and data-flow diversity
- Random generation

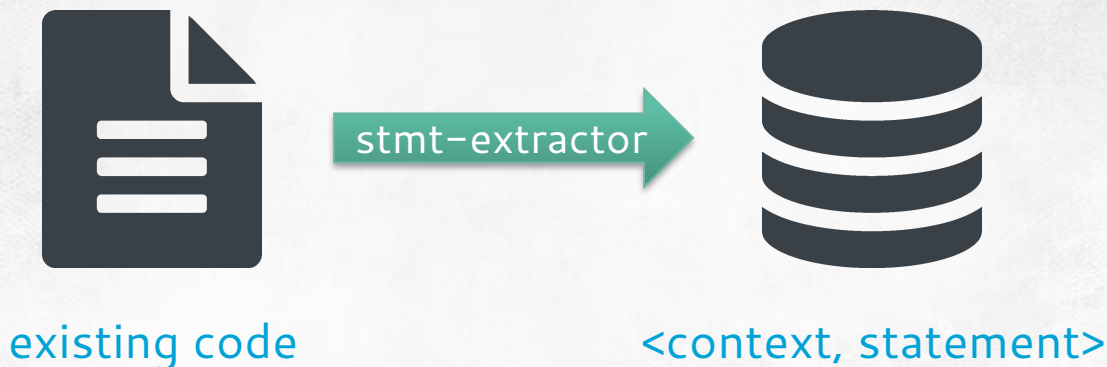


ATHENA

- Better mutation: deletion + injection
 - Generates unlimited and diverse variants
- Guided generation: MCMC sampling
 - Exposes deep compiler bugs

CODE INJECTION

What to Inject?



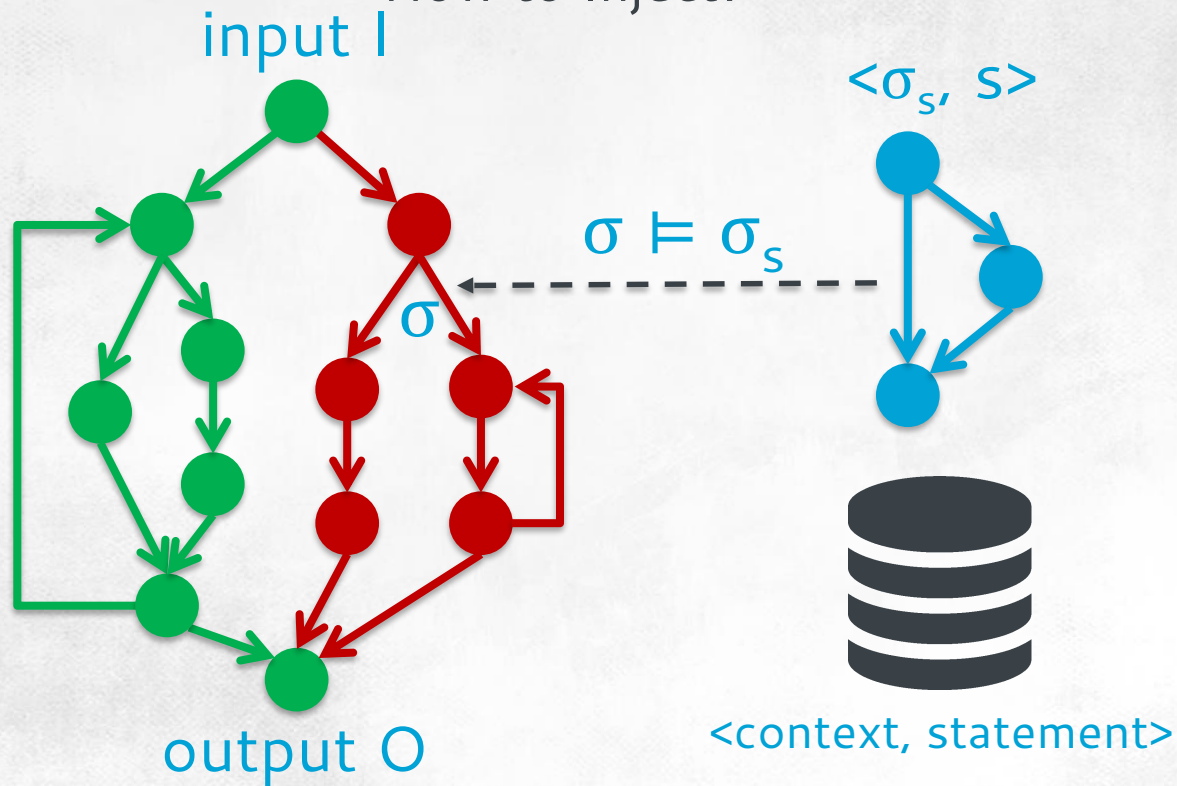
Context: conditions to apply a statement

- Used variables, functions, types, goto labels
- Other properties (e.g., inserted loc must be in a loop)

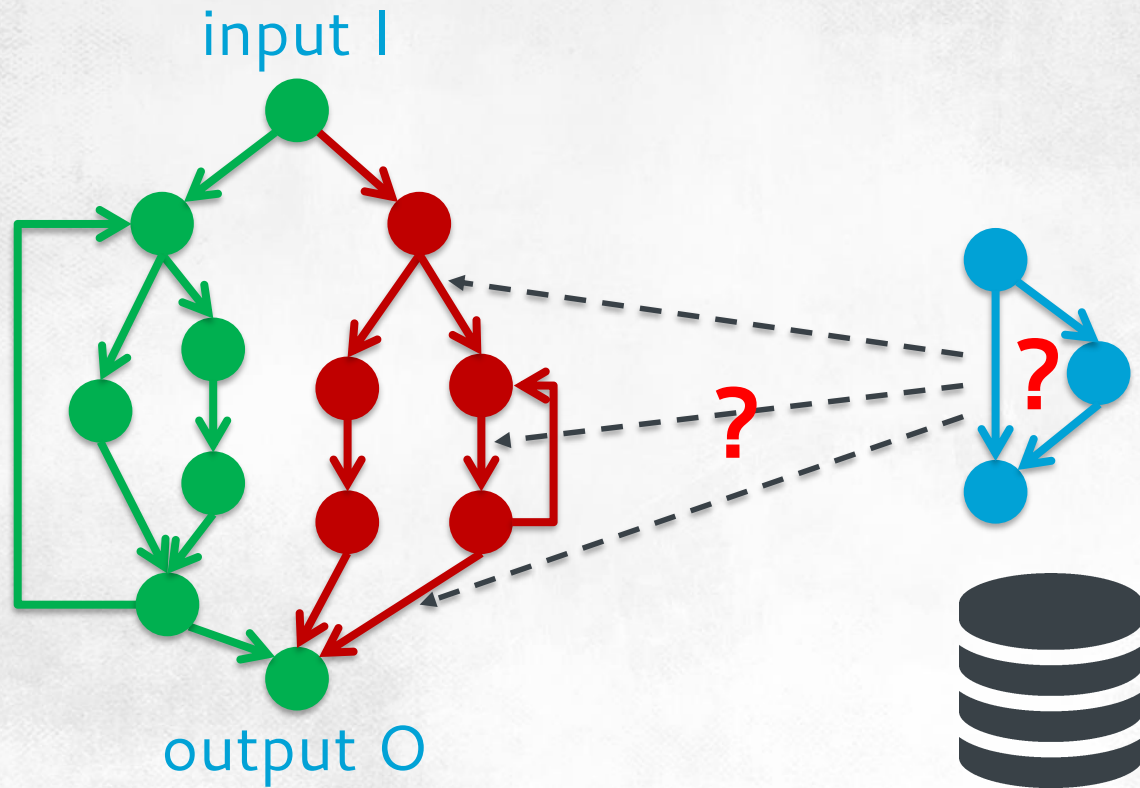


CODE INJECTION

How to Inject?



PROBLEMS



BUG FINDING PROCESS

optimization problem

Goal: generate more **diverse** variants



OBJECTIVE FUNCTION

Program Distance

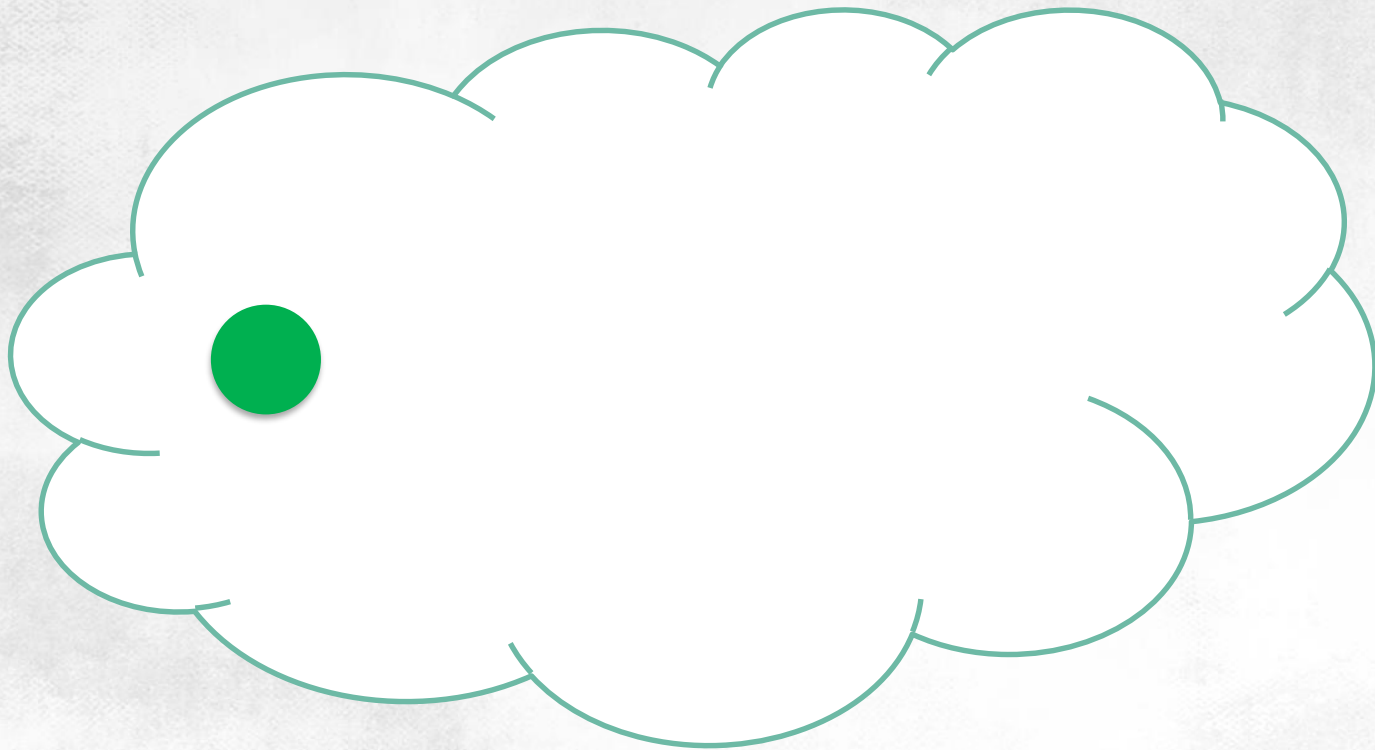
$$\begin{aligned}\Delta(P, Q) = & \alpha * d(P_{Nodes}, Q_{Nodes}) \\ & + \beta * d(P_{Edges}, Q_{Edges}) \\ & - \gamma * |P - Q|\end{aligned}$$

where $d(A, B) = 1 - \frac{|A \cap B|}{|A \cup B|}$ is Jaccard distance



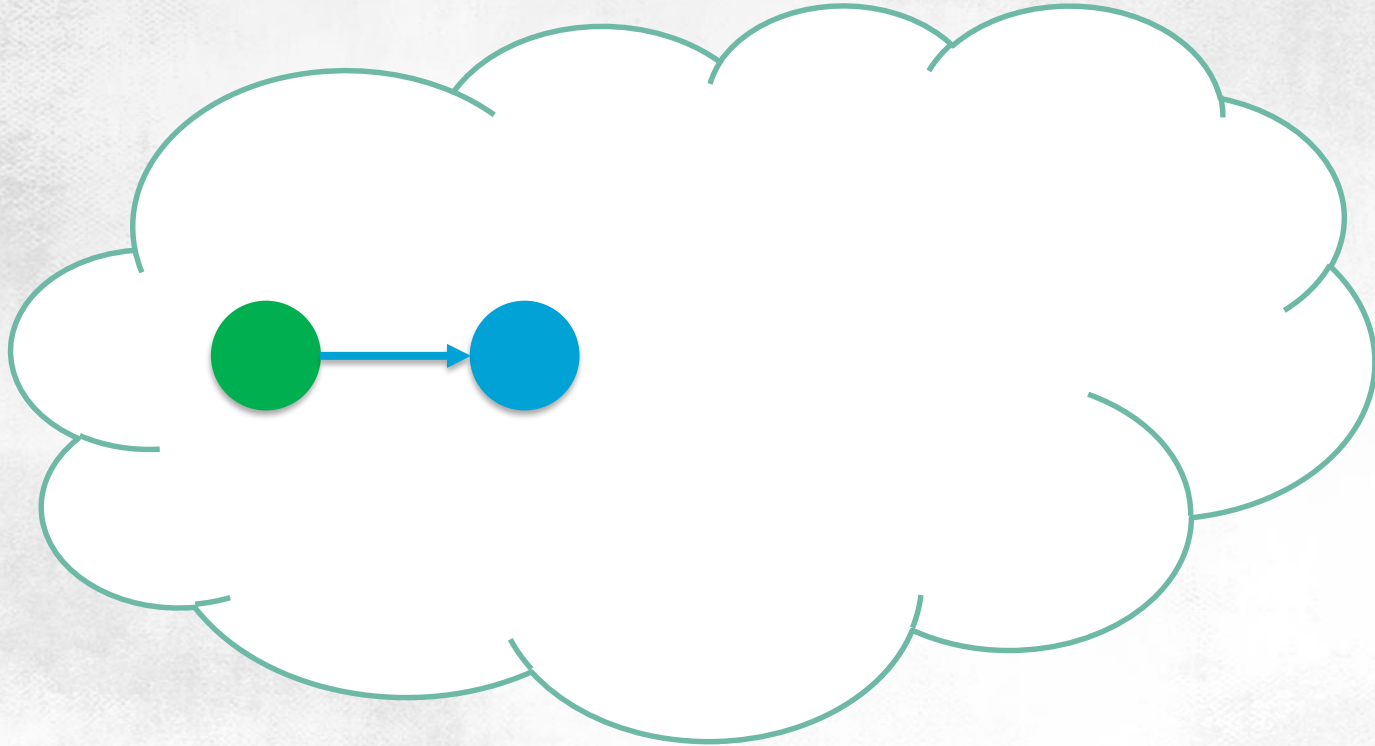
MARKOV CHAIN MONTE CARLO

Sampling High-value EMI Variants



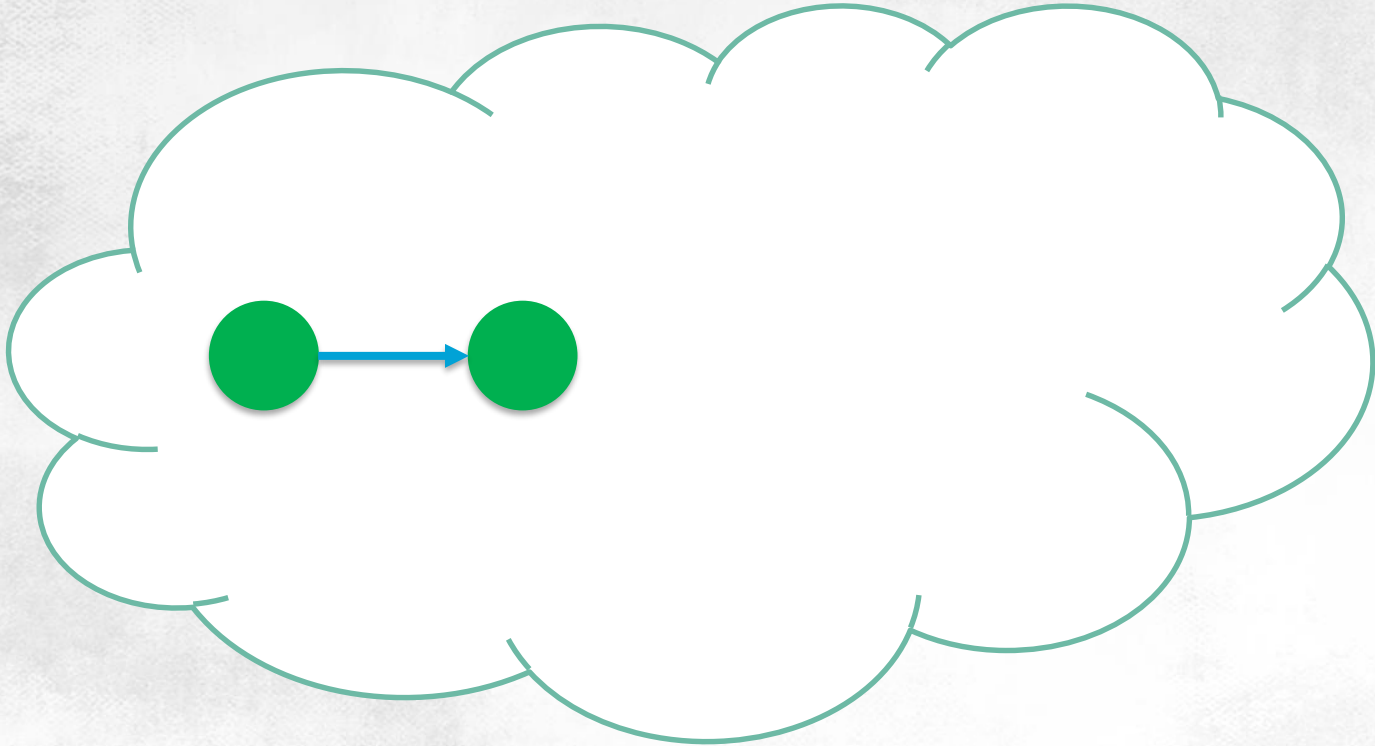
MARKOV CHAIN MONTE CARLO

Sampling High-value EMI Variants



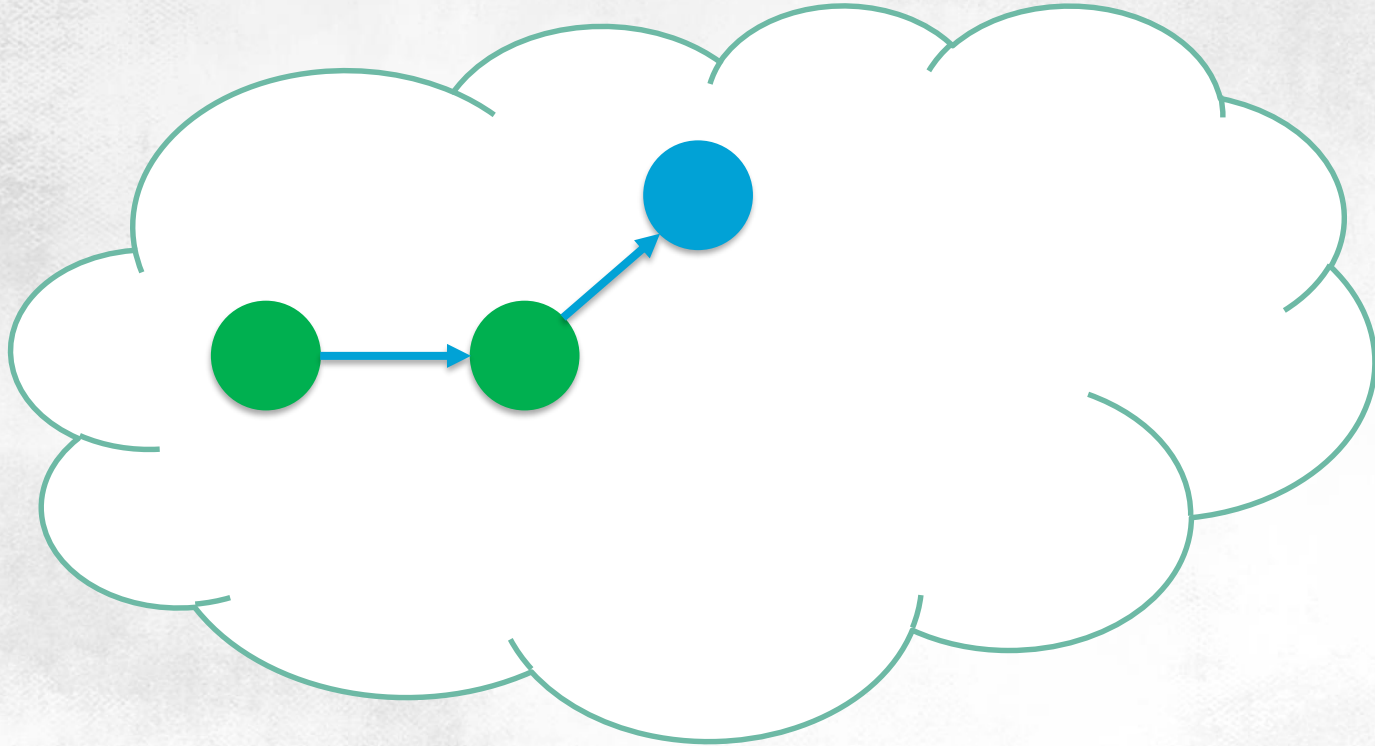
MARKOV CHAIN MONTE CARLO

Sampling High-value EMI Variants



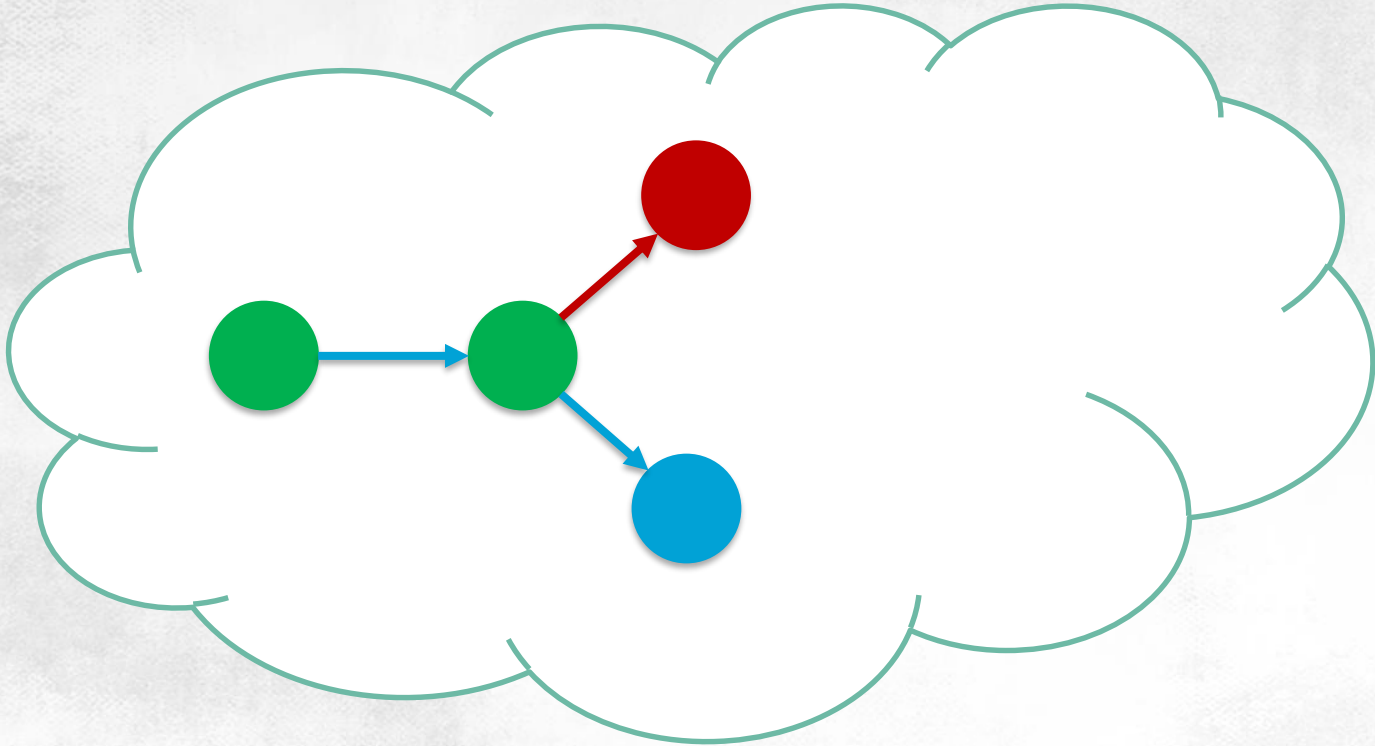
MARKOV CHAIN MONTE CARLO

Sampling High-value EMI Variants



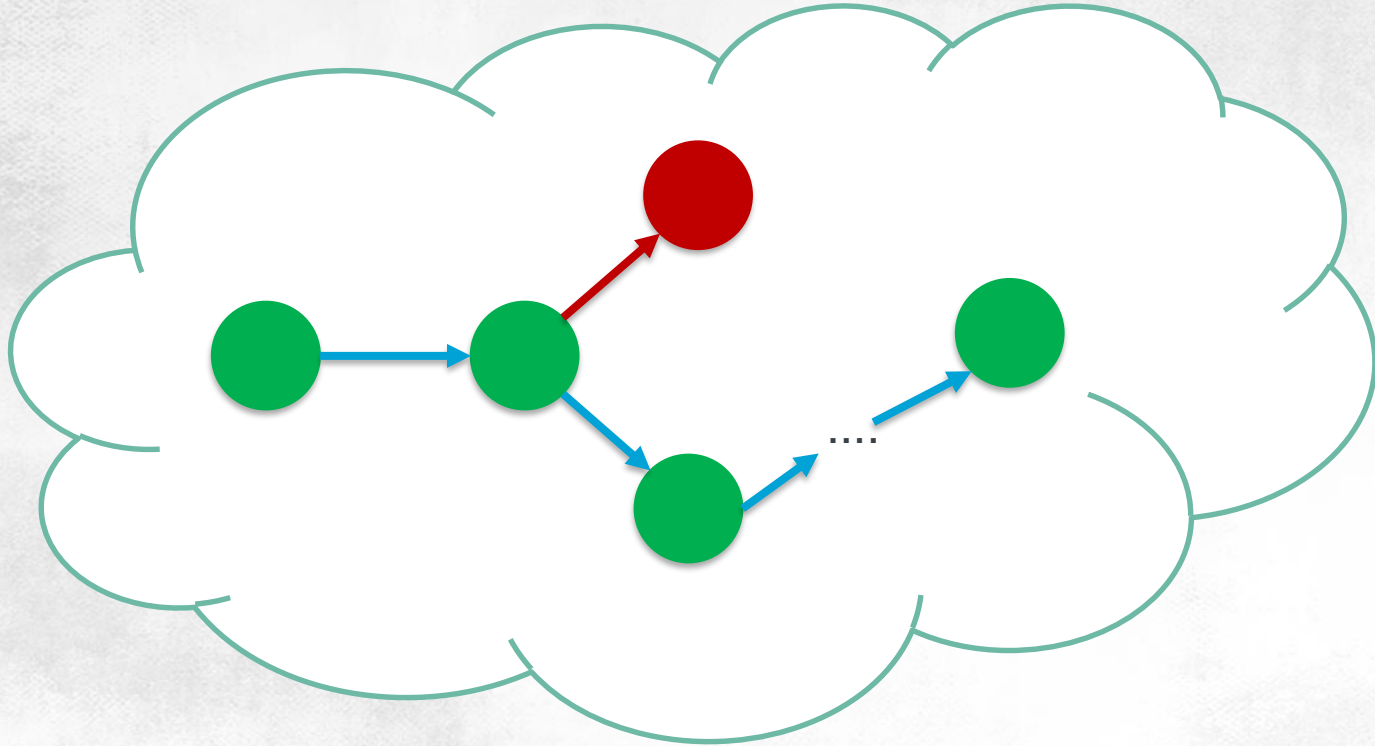
MARKOV CHAIN MONTE CARLO

Sampling High-value EMI Variants



MARKOV CHAIN MONTE CARLO

Sampling High-value EMI Variants



GCC BUG 61383

```
int a, c, d, e = 1, f;
int fn1 () {
    int h;
    for (; d < 1; d = e) {
        h = (f == 0) ? 0 : 1 % f;
        if (f < 1) c = 0;
        else if (h) break;
    }
}
int main () {
    fn1 ();
    return 0;
}
```

>_

```
$ gcc -O0 test.c ; ./a.out
$ gcc -O2 test.c ; ./a.out
Floating point exception (core
dumped)
```

https://gcc.gnu.org/bugzilla/show_bug.cgi?id=61383



GCC BUG 61383

```
int a, c, d, e = 1, f;  
int fn1 () {  
    int h;  
    for (; d < 1; d = e) {  
        h = (f == 0) ? 0 : 1 % f;  
        if (f < 1) c = 0;  
        else c = 1;  
    }  
}  
int main () {  
    fn1 ();  
    return 0;  
}
```



Athena

==DB Entry==
requires_loop
i: int

if (i) break;

```
int a, c, d, e = 1, f;  
int fn1 () {  
    int h;  
    for (; d < 1; d = e) {  
        h = (f == 0) ? 0 : 1 % f;  
        if (f < 1) c = 0;  
        else if (h) break;  
    }  
}  
int main () {  
    fn1 ();  
    return 0;  
}
```



GCC BUG 61383

```
int a, c, d, e = 1, f;  
int fn1 () {  
    int h;  
    for (; d < 1; d = e) {  
        h = (f == 0) ? 0 : 1 % f;  
        if (f < 1) c = 0;  
        else if (h) break;  
    }  
}  
int main () {  
    fn1 ();  
    return 0;  
}
```

PRE: loop invariant

GCC BUG 61383

```
int a, c, d, e = 1, f;  
int fn1 () {  
    int h;  
    int g = 1 % f;  
    for (; d < 1; d = e) {  
        h = (f == 0) ? 0 : g;  
        if (f < 1) c = 0;  
        else if (h) break;  
    }  
}  
int main () {  
    fn1 ();  
    return 0;  
}
```

LIM: hoist (1 % f)

>_

```
$ gcc -O0 test.c ; ./a.out  
$ gcc -O2 test.c ; ./a.out  
Floating point exception (core  
dumped)
```

LIM: Loop Invariant Motion



LLVM BUG 18615

```
int a;  
struct S0 {  
    int f0; int f1; int f2;  
};  
void fn1 () {  
    int b = -1;  
    struct S0 f[1];  
    if (a) {  
        f[0] = f[b];  
    }  
}  
int main () {  
    fn1 ();  
    return 0;  
}
```

>_

\$ clang -O0 test.c

\$ clang -O1 test.c

clang: Assertion failed.

clang: error: Aborted (core dumped)

https://llvm.org/bugs/show_bug.cgi?id=18615



LLVM BUG 18615

```
int a;  
struct SO {  
    int f0; int f1; int f2;  
};  
void fn1 () {  
    int b = -1;  
    struct SO f[1];  
    if (a) {  
        f[0].f0 = b;  
    }  
}  
int main () {  
    fn1 ();  
    return 0;  
}
```



Athena

```
=====DB Entry=====  
g: struct (int x int x int) [1]  
c: int  
-----  
g[0] = g[c];
```

```
int a;  
struct SO {  
    int f0; int f1; int f2;  
};  
void fn1 () {  
    int b = -1;  
    struct SO f[1];  
    if (a) {  
        f[0] = f[b];  
    }  
}  
int main () {  
    fn1 ();  
    return 0;  
}
```



LLVM BUG 18615

```
int a;  
struct S0 {  
    int f0; int f1; int f2;  
};  
void fn1 () {  
    int b = -1;  
    struct S0 f[1];  
    if (a) {  
        f[0] = f[b];  
    }  
}  
int main () {  
    fn1 ();  
    return 0;  
}
```

Assertion Violation:
negative index



https://llvm.org/bugs/show_bug.cgi?id=18615



EVALUATION

- Two machines running in 19 months
- Seed programs: Csmith[1]
 - Hard to reduce real-world projects
- Statement database: seed program
 - Real-world code cannot be inserted into Csmith seeds effectively

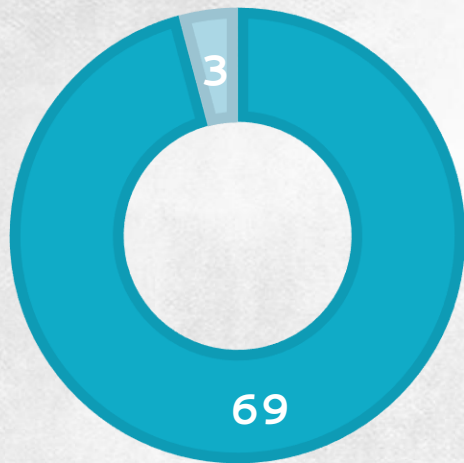


RESULTS

19 months

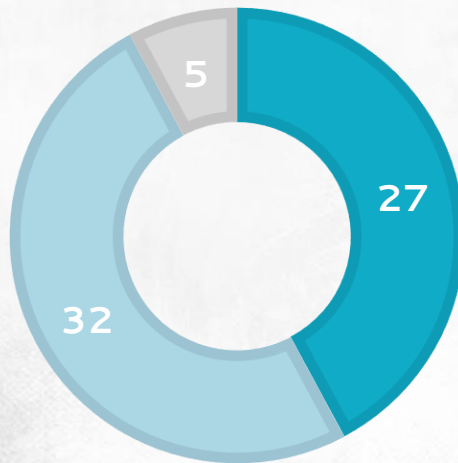
TOTAL BUGS

■ Fixed ■ Confirmed



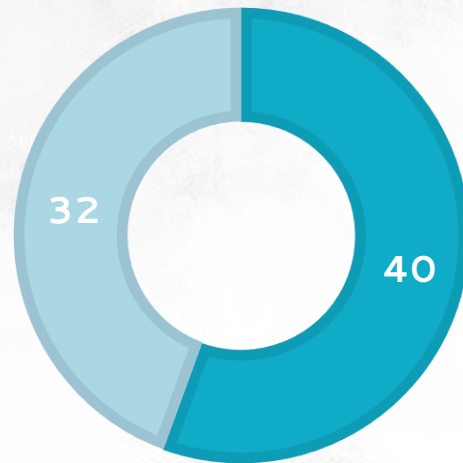
BUG TYPES

■ Wrong ■ Crash ■ Perf



COMPILERS

■ GCC ■ LLVM



IMPORTANCE OF REPORTED BUGS

- Developers **fixed** our bugs (69/72)
- 17/40 GCC bugs are **P1** (highest priority)
- 3 GCC bugs linked to **real-world** projects
 - GCC
 - QtWebKit
 - glibc

ATHENA V.S. ORION

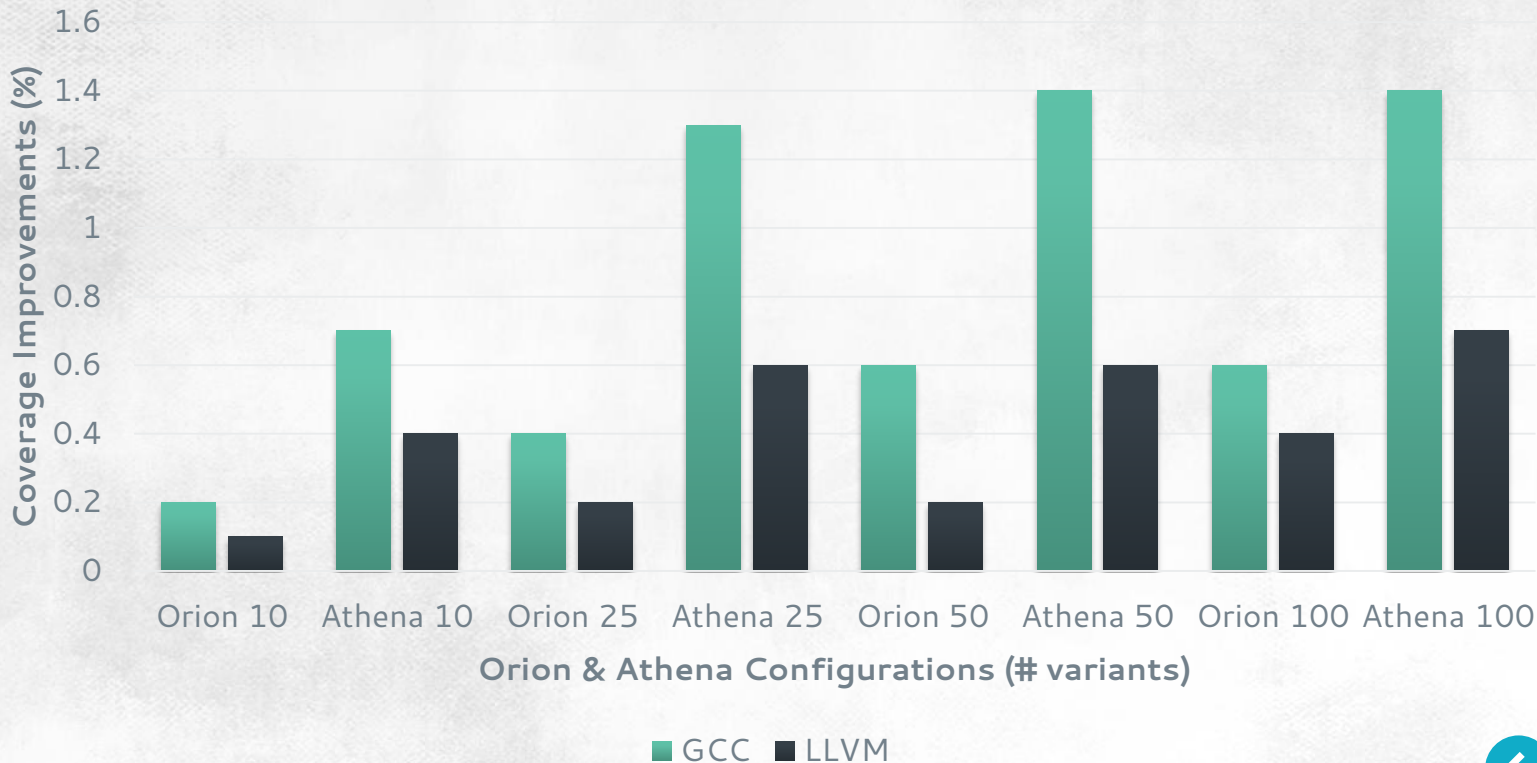
Run Athena and Orion in parallel on 15 bugs in 1 week

Bug ID	Affected Versions	Affected Opt Levels	Seed SLOC	Variant SLOC	Database Rows	Recovered Bugs	Generated Variants
gcc-59903	4.8, 4.9	-O3	4,694	6,238	1,723	14	23,479
gcc-60116	4.8, 4.9	-Os	11,596	11,843	3,092	367	20,082
gcc-60382	4.8, 4.9	-O3	6,151	21,903	1,989	19	21,267
gcc-61383	4.8, 4.9, 4.10	-O2, -O3	3,298	3,567	1,272	106	32,981
gcc-61452	4.8, 4.9, 4.10, 5.0	-O1, -Os	3,308	3,474	885	0	49,158
gcc-61917	4.9, 4.10, 5.0	-O3	11,820	11,226	3,066	2	32,562
gcc-64495	4.8, 4.9, 4.10, 5.0	-O3	2,767	1,951	517	4	45,896
gcc-64663	4.6, 4.7, 4.8, 4.9, 4.10, 5.0	-O1, -Os, -O2, -O3	11,118	12,160	2,875	0	26,626
llvm-20494	3.2, 3.3, 3.4, 3.5	-O2, -O3	8,080	11,009	1,683	2,660	24,588
llvm-20680	3.5, 3.6	-O3	6,250	7,584	1,753	22	23,438
llvm-21512	3.5, 3.6	-O1, -Os, -O2, -O3	8,455	5,087	3,081	988	21,882
llvm-22086	3.5, 3.6	-Os, -O2, -O3	5,220	8,495	1,711	0	29,279
llvm-22338	3.5, 3.6, 3.7	-O2, -O3	2,923	7,197	1,302	13	19,469
llvm-22382	3.2, 3.3, 3.4, 3.5, 3.6, 3.7	-Os, -O2, -O3	4,813	2,147	1,432	0	29,805
llvm-22704	3.6, 3.7	-O1, -Os, -O2, -O3	3,684	23,250	981	12	28,740

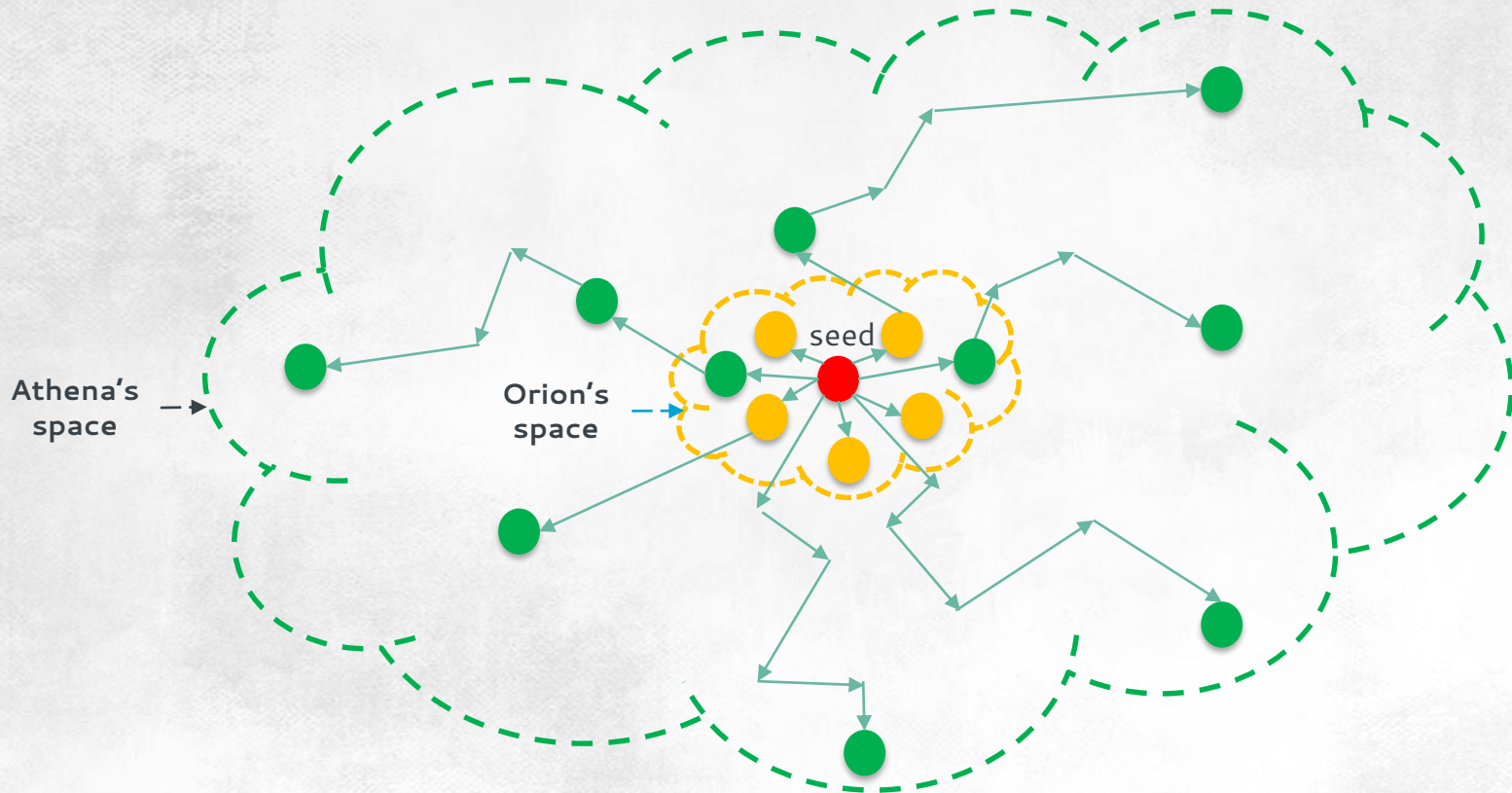


LINE COVERAGE IMPROVEMENT

Baseline: coverage of 100 seeds (GCC 34.9%, LLVM 23.5%)



EMI VARIANT SPACE





**THANK
YOU**



Questions?

REPORTED BUGS

	GCC	LLVM	TOTAL
Fixed	39	30	69
Not-Yet-Fixed	1	2	3
WorksForMe	0	3	3
Duplicate	3	4	7
Invalid	1	0	1
TOTAL	44	39	83