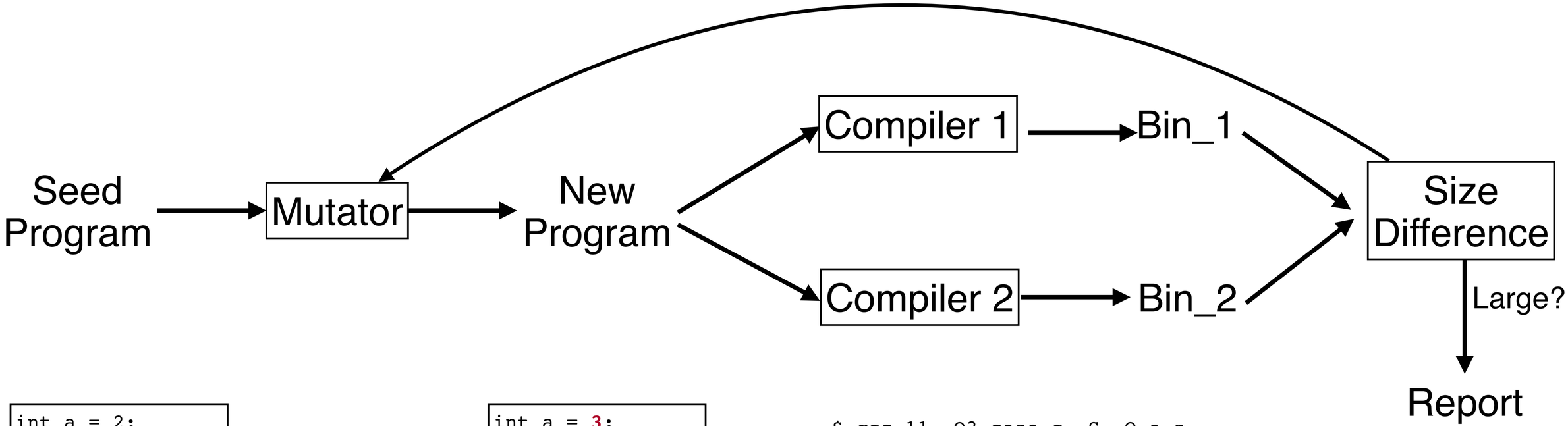
AST Project #4:
Compiler Fuzzing via Guided ValueMutation

Shaohua Li

Project Overview




```
int a = 2;
int b, c, d;
int main() {
    int f = -1;
    if (b)
        c = 0;
    c || (f = 2);
    return 0;
}
```

```
int a = 3;
int b=1, c, d;
int main() {
    int f = 5;
    if (b)
        c = 0;
    c || (f = 2);
    return 0;
}
```

```
$ gcc-11 -O3 case.c -S -O a.s
$ cat a.s|wc -l
20
$
```

```
$ gcc-12 -O3 case.c S —o b.s
$ cat b.s|wc -l
40
$
```

Mutator

Seed programs:

- GCC and Clang C testsuites. (Useful link "Test files")
- All seed programs should be runnable, i.e., containing main() call.

Mutation:

- **Value mutation**: Can be implemented based on pattern matching.
- (Optional) Other mutation: mutate code constructs.

Post-checking:

- check the mutated program against undefined behaviors using AddressSanitizer, UndefinedbehaviorSanitizer.

Compiler Under Test

General Principle: use the same compiler in different versions

- Yes: **gcc-10 -O3** and **gcc-12 -O3**
- Yes: **gcc-10 -O2** and **gcc-12 -O2**
- Yes: **gcc-9 -O2, gcc-10 -O2,** and **gcc-12 -O2**
- No: **gcc-12 -O2** and **gcc-12 -O3**
- No: **clang-16 -O1** and **gcc-12 -O1**

Size difference:

- **The only interesting case**: Binary compiled by the latest version has significantly larger size than old version
- **Measurement of size:** the number of instructions in the compiled assembly code.

Feedback from Size Difference

- No feedback: randomly mutate
- (Optional)With feedback: mutate according to the feedback from the size difference of the current mutated program.

Grading

Final submission (assume 100 pts):

- (60 pts) Report
- (20 pts) Source code and detailed instructions to run your tool.
- (20 pts) at least **10 source files** with the most significant difference you have found. Please provide both the source files and two compilers that causes significant differences.