Code   Paper

Verifying the Option Type with Rely-Guarantee Reasoning

James Yoo, Michael D. Ernst, René Just
University of Washington

The Problem with the Optional Type

```
Optional<File> file = ...
...
file().get(); ✗
...
if (file.isPresent()) {
  file.get(); ✓
}
```

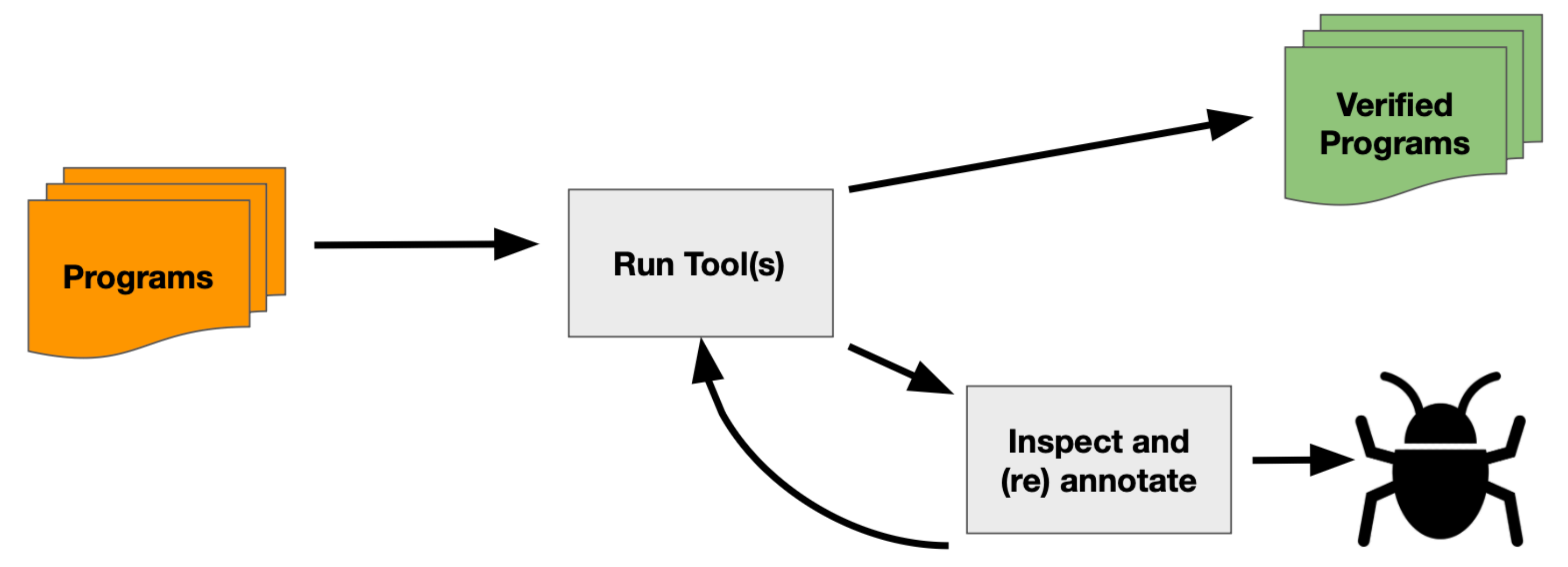
Unsafe! Absent option value leads to **run-time** exception

Safe! Absent option value never accessed

- Unenforced presence/absence checks
- Ugly, inefficient, and redundant code

Evaluation

Across 1M SLOC open-source Java programs



Optional Checker
- 🏆 13 real-world defects
- 🏆 **93% precision, 100% recall**
- **6 programmer-written annotations**

Prior Tools
- 69% precision, 85% recall (IntelliJ)
- 🏆 **0 programmer-written annotations**
- SpotBugs and Error Prone missed all defects

```
private Optional<String> prefix = ...

Function<..., ...> build() {
  ...
  if (prefix.isPresent()) {
    return m -> prefix.get();
  }
  ...
}
```

```
Type Error!..>
  found   : @MaybePresent
  required: @Present
```

```
1: fn = builder.build();
2: builder
     .setPrefix(
       Optional.empty());
3: fn.apply(...);
```

Solution The Optional Checker

Eliminate run-time errors rooted in accessing **absent option values** with a **type system** that **explicitly models present** and **absent** option values

$$\begin{tikzcd}
\text{@MaybePresent} & \text{@UnknownNonEmpty} \\
\text{@Present} \arrow[u] & \text{@NonEmpty} \arrow[u]
\end{tikzcd}$$

```
get(@Present Optional<T> this);
orElse(@Present Optional<T> this);
```

```
@MaybePresent
Optional<File> file = ...
...
file().get();
...
if (file.isPresent()) {
  file.get();
}
@Present
```

```
Type Error!
  found   : @MaybePresent
  required: @Present
```

Key Idea Partial Rely-Guarantee Reasoning

Verify only the parts of the program that are **relevant to your guarantee**

userIds	@NonEmpty List<Integer>

```
// userIds is non-empty
List<Integer> userIds = ...
```

userIds	@NonEmpty List<Integer>
maxUserId	@Present Optional

```
Optional<Integer> maxUserId =
    userIds.stream()
    .max(Integer::compareTo);


                    maxUserId.get();
```

CHECKER framework   PLSE   W PAUL G. ALLEN SCHOOL OF COMPUTER SCIENCE & ENGINEERING