KOTLIN FUZZER

Михаил Кольцов

Руководитель: Марат Ахин

СТРУКТУРА ДОКЛАДА

- о чём задача
- как решал
- какие сложности
- результаты

ЗАДАЧА

- язык Kotlin - много функций
- компилятор сложный
- хотим генерировать файлы, на которых kotlinc упадёт с ошибкой во время компиляции

МОТИВАЦИЯ

- JavaScript и C успешно тестируют
- Kotlin развивается
- повысим надёжность компилятора

ВДОХНОВЛЯЮЩИЙ ПРИМЕР: KT-11902

```
class A {
    open inner class AB
}

fun A.foo() {
    class FooC : A.AB()
}

fun main(args: Array<String>) {
    A().foo()
}
```

РАССМОТРЕННЫЕ ПУТИ РЕШЕНИЯ

- Java PathFinder, jFuzz, ...
- Quickcheck, ...
- реализовать идею из статьи
- свой фаззер

ПОЧЕМУ ПИШЕМ СВОЙ ФАЗЗЕР

- symbolic execution долго работает
- соответствие грамматике не гарантирует компилируемость

```
fun f() {
  return g()
}
```

- конфигурирование существующих инструментов занимает время

СХЕМА РЕШЕНИЯ

1. Берём файл с существующим кодом
2. Парсим, получаем дерево
3. Вставляем новый код согласно шаблонам
4. Реконструируем код по дереву
5. Подаём компилятору

СХЕМА РЕШЕНИЯ

- где брать код: compiler/testData (9033 файла с расширением .kt)
- где взять грамматику: из документации
- как парсить: генератор синтаксических анализаторов ANTLR4
- шаблоны: придумывать головой

ЧТО НЕ ТАК С ГРАММАТИКОЙ ИЗ ДОКУМЕНТАЦИИ

Код

```
fun f() {
    return g()
}
```

Фрагмент грамматики

```
alpha{beta} denotes a nonempty beta-separated list of
alpha's.

valueArguments
  : "(" (SimpleName "=")? "*"? expression{","} ")"
  ;
```

ЧТО НЕ ТАК С ГРАММАТИКОЙ ИЗ ДОКУМЕНТАЦИИ

```
callSuffix
  : typeArguments? valueArguments annotatedLambda
  : typeArguments annotatedLambda
  ;

annotatedLambda
  : ("@" unescapedAnnotation)* labelDefinition?
    functionLiteral
  ;

functionLiteral
  : "{" statements "}"
  : "{" lambdaParameter{","} "->" statements "}"
  ;
```

ПАРСЕР

- грамматику писал долго
- отлаживать грамматику сложно
- получил грамматику для ANTLR4, которая без ошибок парсит ~5000 файлов

ГДЕ ПАРСЕР ОШИБАЕТСЯ

- некоторые файлы не должны компилироваться (код на java, тесты на ошибки компиляции)
- в коде бывает информация для компилятора
- переводы строк не везде поддерживаю
- неправильно обрабатываю некоторые выражения (when, property)

РЕЗУЛЬТАТЫ

- изучил подходы к фаззингу
- познакомился с языком Kotlin
- выбрал подходящий метод решения имеющейся задачи
- написал работающую грамматику языка

СПАСИБО ЗА ВНИМАНИЕ!