Построение регулярной аппроксимации встроенных языков

Студент: Хабибуллин Марат Рафисович
Руководитель: Григорьев Семён Вячеславович
СПбАУ НОЦНТ РАН, 2015

Введение

```
Entries getEntries() {
    String usual = "Hello!";
    String sql = "SELECT * FROM ";
    if(cond)
        sql = sql + "Table_1";
    else
        sql = sql + "Table_2";
    return db.exec(sql);
}
```

Статический анализ встроенных языков для:

- сред разработки
- реинжиниринга

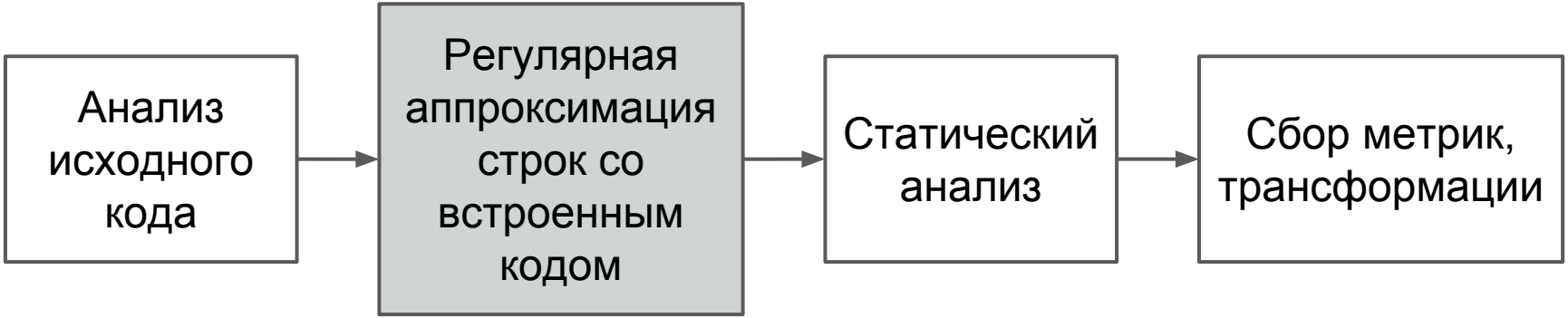
Реинжиниринг

1. Сбор статистики по коду на встроенном языке
   a. формальные метрики
   b. специфичные метрики (используемость таблиц в SQL запросах)
2. Автоматизированная трансформация

Существующие решения

- **PhpStorm** - не поддерживает строковые операции
- **IntelliLang** - не умеет находить встроенный код сам
- **Alvor** - SQL внутри Java
- **Varis** - JavaScript, CSS, HTML в Php
- **Java String Analyzer** - любые языки внутри Java (плохо расширяем)
- **Php String Analyzer** - любые языки внутри Php (плохо расширяем)

Подход к решению



Fang Yu, Muath Alkhalaf,Tevfik Bultan,Oscar H. Ibarra.

Automata-based symbolic string analysis for vulnerability detection (2014)

Цель и задачи

Цель - построение регулярной аппроксимации строковых выражений без привязки к основному языку

Задачи:

1. Реализовать алгоритм построения аппроксимации
2. Сделать решение независимым от основного языка
3. Интегрировать результат в YaccConstructor
4. Реализовать для конкретного языка

Схема решения

1. Поиск строк со встроенным кодом
2. Построение графа потока управления
3. Конвертиция в обобщённый графа потока управления
4. Построение конечного автомата по обощённому графу

Поиск строчек с кодом

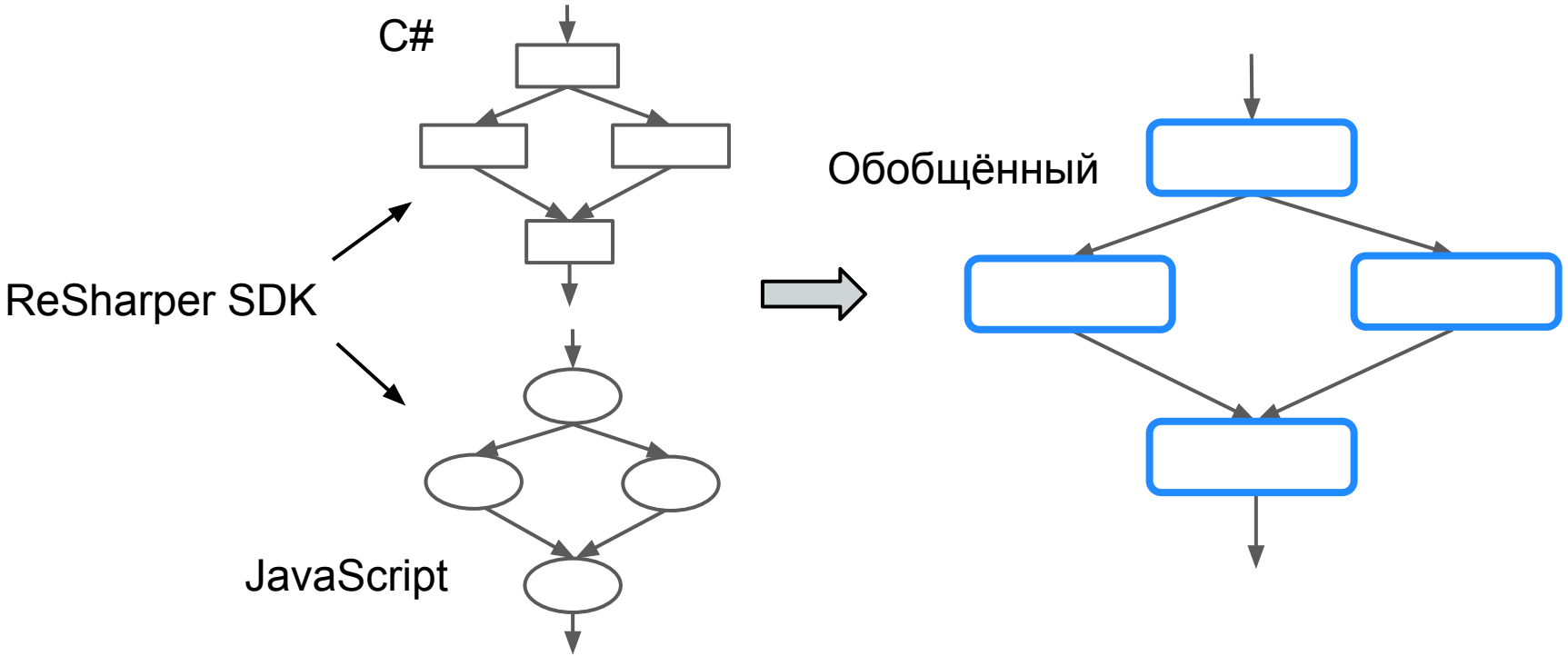
```
Entries getEntries() {
    String usual = "Hello!";
    String sql = "SELECT * FROM ";
    if(cond)
        sql = sql + "Table_1";
    else
        sql = sql + "Table_2";
    return db.exec(sql);
}
```

Тут нет кода

Эта переменная содержит код

Ищем функции с таким именем

Построение множества значений



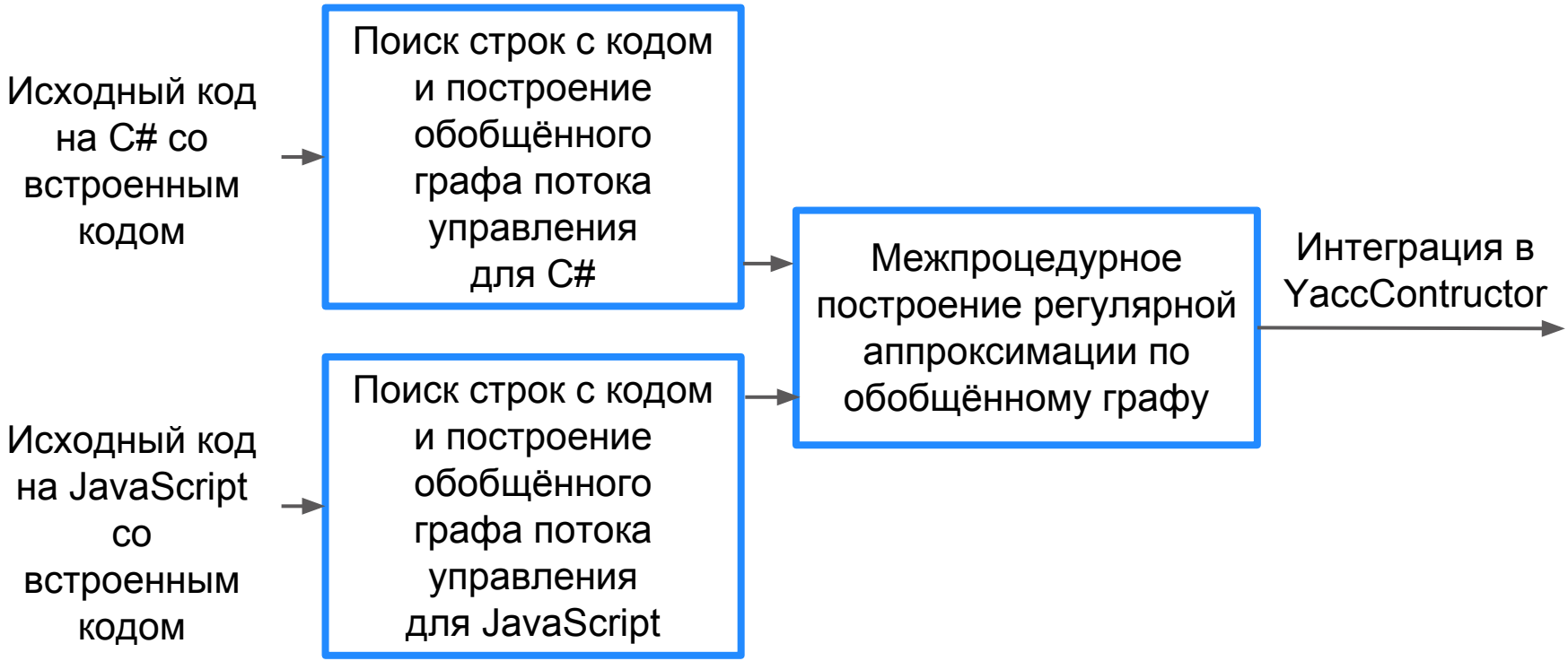
Построение конечного автомата

Строковый литерал	⇔	Автомат, который принимает только этот литерал
Конкатенация и замена в строках	⇔	Конкатенация и замена в автоматах
Условный оператор	⇔	Объединение автоматов
Цикл	⇔	Расширение автоматов
Пользовательский ввод	⇔	Автомат, который принимает любые строки

Обработка функций

- Межпроцедурное построение аппроксимации (рекурсивный запуск алгоритма)
- Хвостовая рекурсия - предобработка графа потока управления (превращение в цикл)
- Нехвостовая рекурсия - не поддерживается

Итоговая архитектура приложения



Результаты

1. Реализовано построение регулярной аппроксимации
   a. Поддержаны основные операции над строками и основные операторы потока управления
   b. Поддержаны нерекурсивные функции и хвостовая рекурсия (межпроцедурная аппроксимация)
   c. Реализован оператор расширения автоматов для поддержки циклов
2. Реализован обобщённый граф потока управления на основе которого строится аппроксимация
3. Интегрировано в YaccConstructor
4. Реализовано построение аппроксимации для C# и JavaScript