



# Генератор абстрактных лексических анализаторов

**Автор:** Полубелова Марина Игоревна

**Научный руководитель:** магистр информационных технологий  
С.В. Григорьев

Санкт-Петербургский государственный университет  
Математико-Механический факультет  
Кафедра системного программирования

22 мая 2014г.

- Реинжиниринг программного обеспечения
  - ▶ Анализ и трансформация систем, код которых содержит встроенные языки
- Поддержка встроенных языков в IDE
  - ▶ Статический поиск ошибок
  - ▶ Подсветка синтаксиса
  - ▶ Рефакторинги

## Пример встроенного языка

```
private void Go(int cond)
{
    string tableName;
    if (cond > 0)
    {
        tableName = "1";
    }
    else
    {
        tableName = " 2";
    }
    Program.ExecuteImmediate("select x from y " + "where a ++ b > 2");
    Program.ExecuteImmediate("select x from name" + tableName);
}
```

## Пример встроенного языка

```
private void Go(int cond)
{
    string tableName;
    if (cond > 0)
    {
        tableName = "1";
    }
    else
    {
        tableName = "2";
    }
    Program.ExecuteImmediate("select x from y " + "where a ++ b > 2");
    Program.ExecuteImmediate("select x from name" + tableName);
}
```

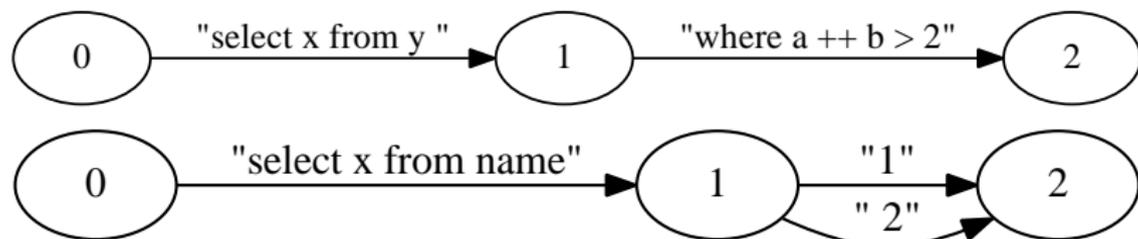
## Пример встроенного языка

```
private void Go(int cond)
{
    string tableName;
    if (cond > 0)
    {
        tableName = "1";
    }
    else
    {
        tableName = "2";
    }
    Program.ExecuteImmediate("select x from y " + "where a ++ b > 2");
    Program.ExecuteImmediate("select x from name" + tableName);
}
```



## Пример встроенного языка

```
private void Go(int cond)
{
    string tableName;
    if (cond > 0)
    {
        tableName = "1";
    }
    else
    {
        tableName = "2";
    }
    Program.ExecuteImmediate("select x from y " + "where a ++ b > 2");
    Program.ExecuteImmediate("select x from name" + tableName);
}
```



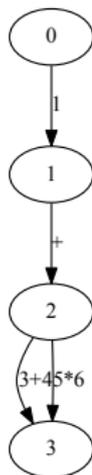
# Абстрактный лексический анализ

Является необходимым шагом при обработке встроенных языков.  
Нужен для токенизации входного графа.

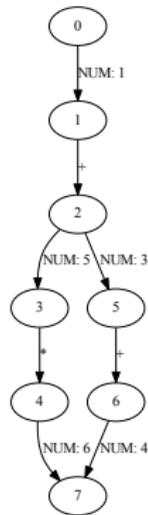
- На вход анализатору подается граф, являющийся представлением динамически формируемого выражения
- На выходе получаем также граф, каждое ребро которого содержит токен
- Результирующий граф пригоден для дальнейшего синтаксического анализа

# Пример

“Вход”



“Лексический анализ”



Выражения: “1 + 3 + 4” и “1 + 5 \* 6”

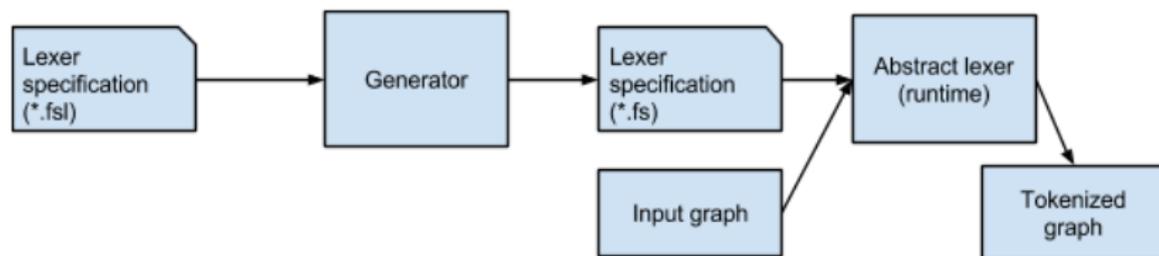
# Обзор существующих решений и аналогов

- Java String Analyzer
  - ▶ строковое выражение аппроксимируют регулярной грамматикой
  - ▶ нет лексического анализа
- PHP String Analyzer
  - ▶ строковое выражение аппроксимируют контекстно-свободной грамматикой
  - ▶ нет лексического анализа
- Alvor
  - ▶ плагин к Eclipse для проверки встроенного в Java SQL
  - ▶ проводят абстрактный лексический анализ
- статья Kyung-Goo Doh, Hyunha Kim, David A. Schmidt
  - ▶ статическая валидация динамически генерируемого HTML в PHP
  - ▶ проводят абстрактный синтаксический анализ
- Курсовая работа Екатерины Вербицкой

# Постановка задачи

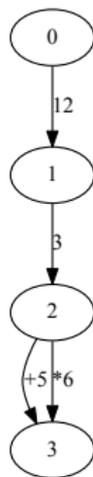
- Доработать генератор абстрактных лексических анализаторов
  - ▶ Поддержка “рваных” токенов
  - ▶ Сохранять привязку частей динамически формируемого выражения к исходному коду
  - ▶ Реализовать привязку лексических единиц внутри каждой части
- Сравнить полученный инструмент с его аналогами

# Структура и принцип работы генератора

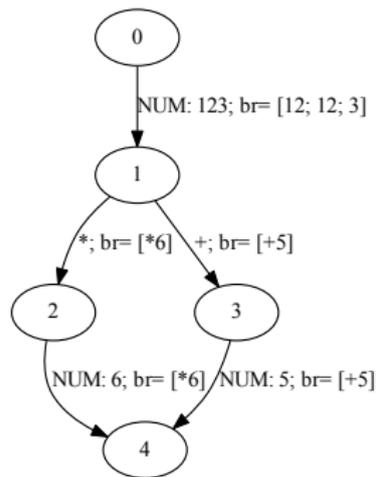


# Сохранение привязки к исходному коду

“Вход”



“Лексический анализ”



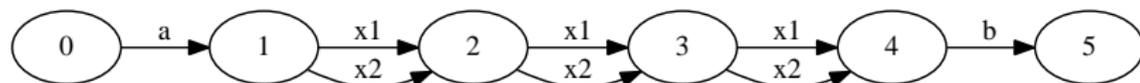
Выражения: “123 + 5” и “123 \* 6”

## Пример работы

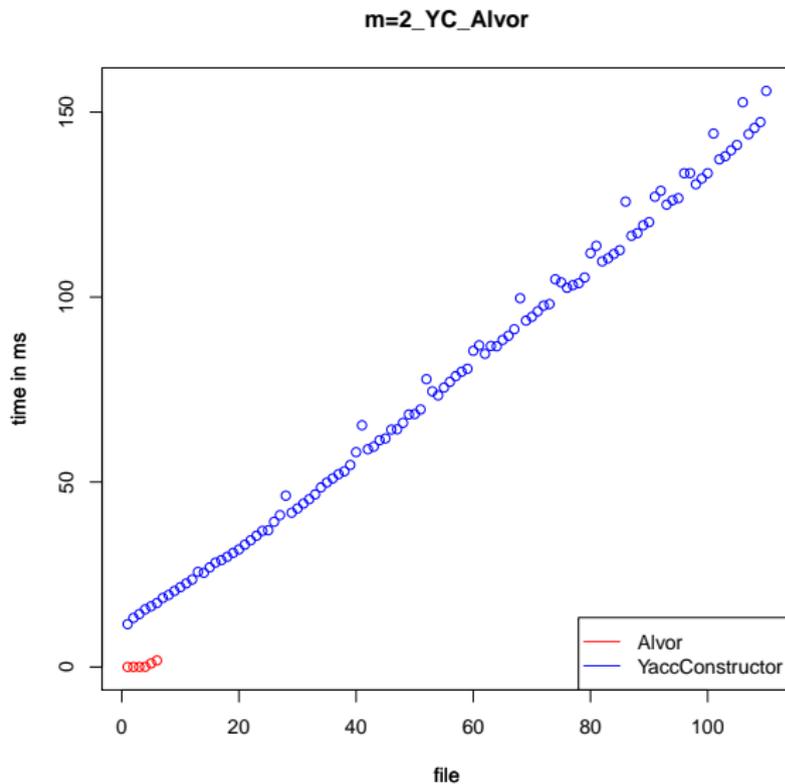
```
private void Go(int cond)
{
    string tableName;
    if (cond > 0)
    {
        tableName = "1";
    }
    else
    {
        tableName = " 2";
    }
    Program.ExecuteImmediate("select x from y " + "where a ++ b > 2");
    Program.ExecuteImmediate("select x from name" + tableName);
}
```

# Генерация тестов

Набор тестов параметризуется кратностью ребра ( $m = 2$ ) и длиной цепочкой ( $n = 3$ )



# Сравнение производительности



# Результаты

- Реализована структура, хранящая информацию о позиции токена в исходном коде
- Осуществлена корректная передача координат токена к ReSharper
- Изучены аналоги лексических анализаторов
- Реализован инструмент для тестирования лексеров YaccConstructor и Alvor
- Проведено сравнение лексера инструментов YaccConstructor и Alvor на одинаковых входных данных
- Результаты работы представлены на конференции “Технологии Microsoft в теории и практике программирования”
- Принята статья “Инструментальная поддержка встроенных языков в интегрированных средах разработки” на семинар по наукоёмкому ПО.