



# Доработка алгоритма синтаксического анализа регулярной аппроксимации кода на встроенных языках YaccConstructor

**Автор:** Екатерина Вербицкая

Лаборатория языковых инструментов JetBrains  
Санкт-Петербургский государственный университет  
Математико-механический факультет

10 сентября 2015г.

- Проект для исследований в области лексического и синтаксического анализа
- Открытый исходный код
  - ▶ <https://github.com/YaccConstructor/YaccConstructor>
- Основной язык разработки — F#

- Синтаксический анализ **всех строк** из регулярного **множества**
- Вход — не одна строка, а множество (возможно бесконечное)
- Множество описано конечным автоматом

# Сфера применимости алгоритма

Встроенный код

```
string res = "";  
for(i = 0; i < 1; i++)  
    res = "()" + res;  
use(res);
```

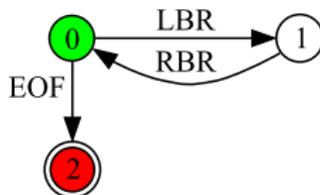
Возможные значения

{ "", "()", "()()", ..., "()"<sup>1</sup> }

Аппроксимация

("()")\*

Соответствующий КА



## Задача: ослабить ограничения на входной автомат

- Детерминированный автомат без  $\epsilon$ -переходов  $\Rightarrow$  произвольный конечный автомат
- Несложная задача на семестр

## Задача: диагностика и сообщение об ошибках

- Предлагается 2 подхода
- На диплом: реализовать и сравнить оба подхода
- Можно разделить на 2 человек
- Результаты можно будет опубликовать

# Требования к знаниям и навыкам

- Знакомство с функциональным программированием
  - ▶ Знание F# или OCaml является преимуществом
- Базовые знания в области синтаксического анализа

- Почта: `kajigor@gmail.com`
- Исходный код YaccConstructor:  
`https://github.com/YaccConstructor/YaccConstructor`
- Google+ сообщество: `https://goo.gl/DuPWkM`
- Больше задач на GitHub:  
`https://github.com/YaccConstructor/YaccConstructor/issues`