



Реализация и оценка эффективности алгоритма обобщенного синтаксического анализа с уменьшенной активностью стека

Автор: Дмитрий Ковалев, 344 группа

Научный руководитель: магистр информационных технологий,
ст.пр. Григорьев С.В.

Санкт-Петербургский государственный университет
Кафедра системного программирования

08 сентября 2016г.

Неоднозначные КС-грамматики

- Области применения
 - ▶ Синтаксический анализ динамически формируемого кода
 - ▶ Биоинформатика
- Алгоритмы
 - ▶ CYK
 - ▶ Earley
 - ▶ Generalized LR (GLR)
 - ▶ RNLGR
- Недостаток — скорость

Reduction Incorporated GLR (RIGLR)

- Сводит использование стека к обработке вложенной рекурсии

$$A \Rightarrow \dots \Rightarrow \alpha A \beta, \alpha, \beta \neq \epsilon$$

- Позволяет с меньшими затратами анализировать регулярную часть языка
- Scott, E. and Johnstone, A. "Generalized Bottom Up Parsers With Reduced Stack Activity", Computer Journal

YaccConstructor (YC)

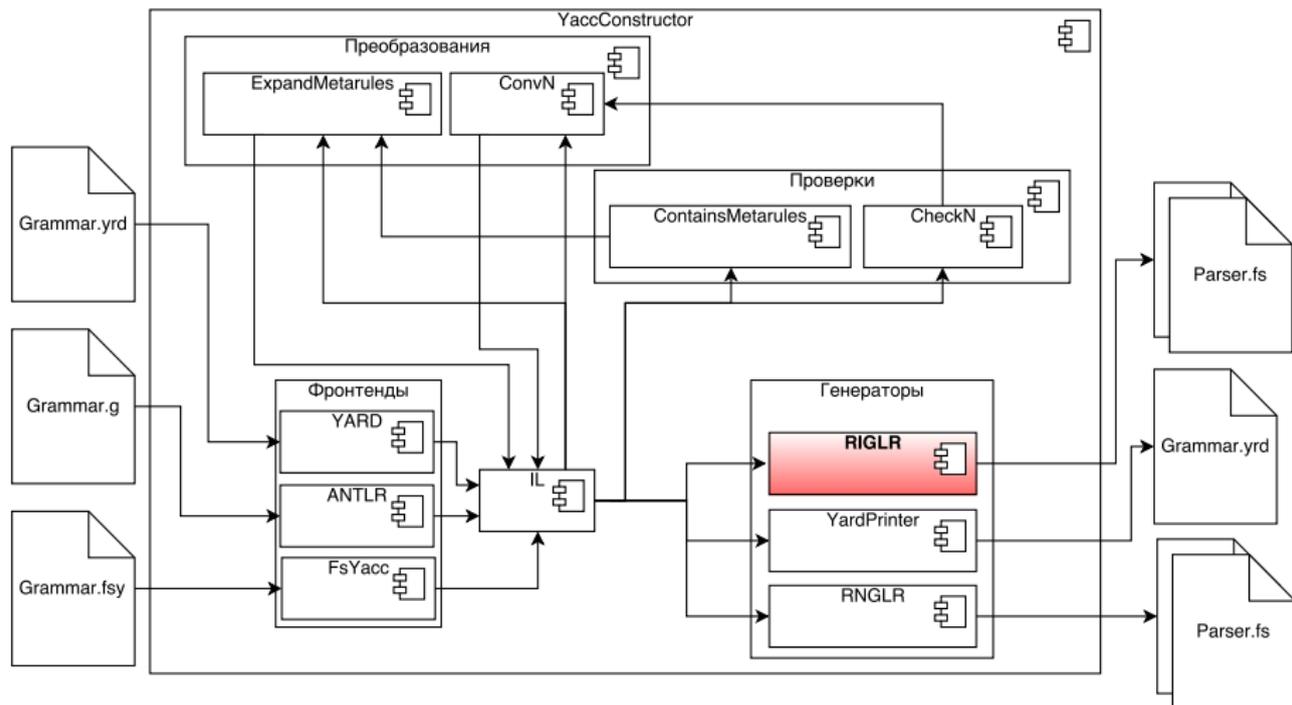
- Проект лаборатории языковых инструментов JetBrains, направленный на исследования в области формальных языков, лексического и синтаксического анализов
- .NET, F#
- Синтаксический анализ динамически формируемого кода на основе RNLGR

Целью данной работы является увеличение производительности алгоритма синтаксического анализа динамически формируемого кода

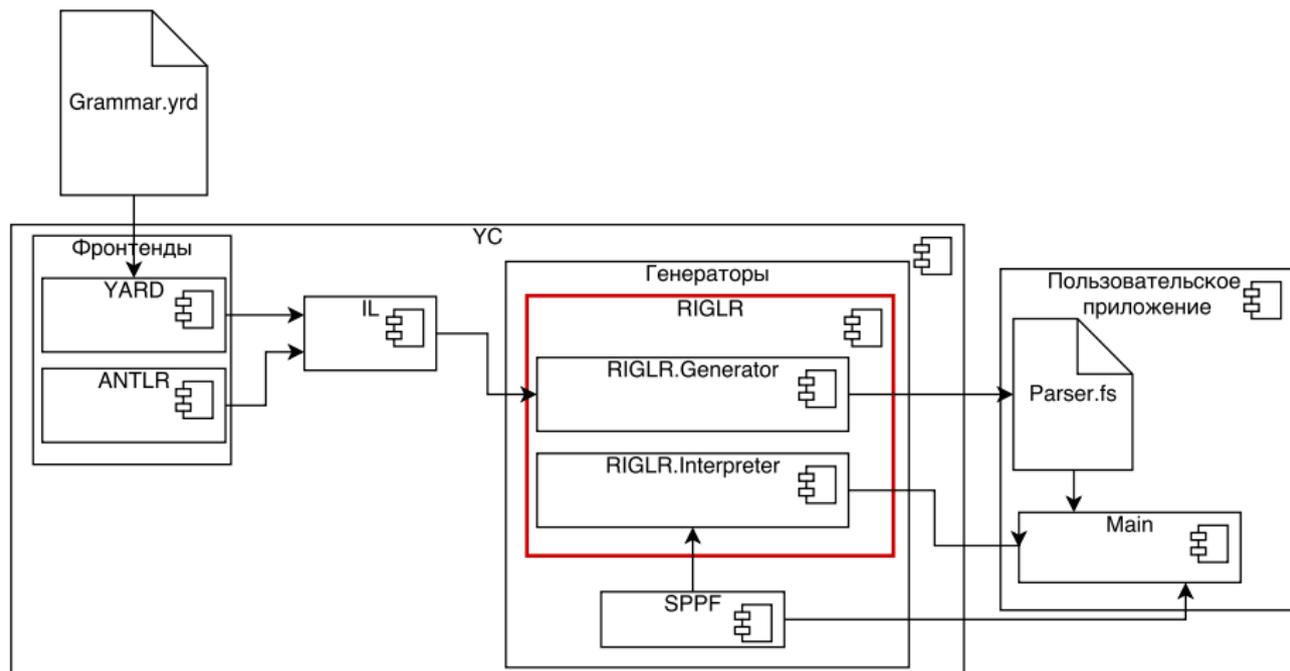
Задачи:

- Изучить RIGLR-алгоритм синтаксического анализа
- Реализовать генератор синтаксических анализаторов, основанный на RIGLR-алгоритме
- Провести сравнение производительности анализаторов, созданных на основе RNLGR и RIGLR

Архитектура платформы



Архитектура RIGLR-модуля



- Модуль генератора
 - ▶ Реализация автоматов, необходимых для создания управляющих таблиц (QuickGraph)
 - ▶ Алгоритм удаления вложенной рекурсии ¹
 - ▶ Генерация файла, содержащего таблицы и дополнительную информацию для парсера
- Модуль интерпретатора
 - ▶ Основная логика RIGLR-анализатора

¹Marcella Anselmo, Dora Giammarresi, and Stefano Varricchio. 2002. Finite automata and non-self-embedding grammars.

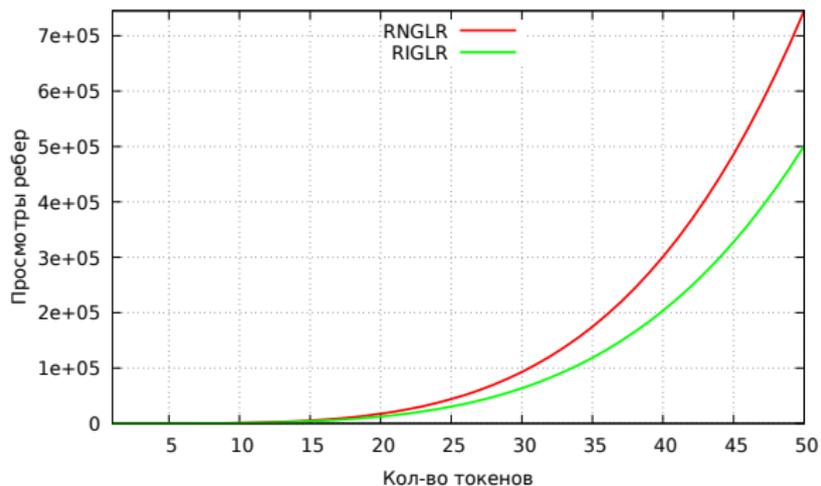
- Сильно неоднозначная грамматика, инициирует худший случай работы

$$\begin{aligned} \textit{start} &::= s \\ s &::= s s s \mid s s \mid A \end{aligned}$$

- Характеристики компьютера
 - ▶ Процессор: Intel(R) Core(TM) i7-4790 CPU 3.60GHz, 4 Core(s)
 - ▶ Оперативная память: 16.0 GB DDR3

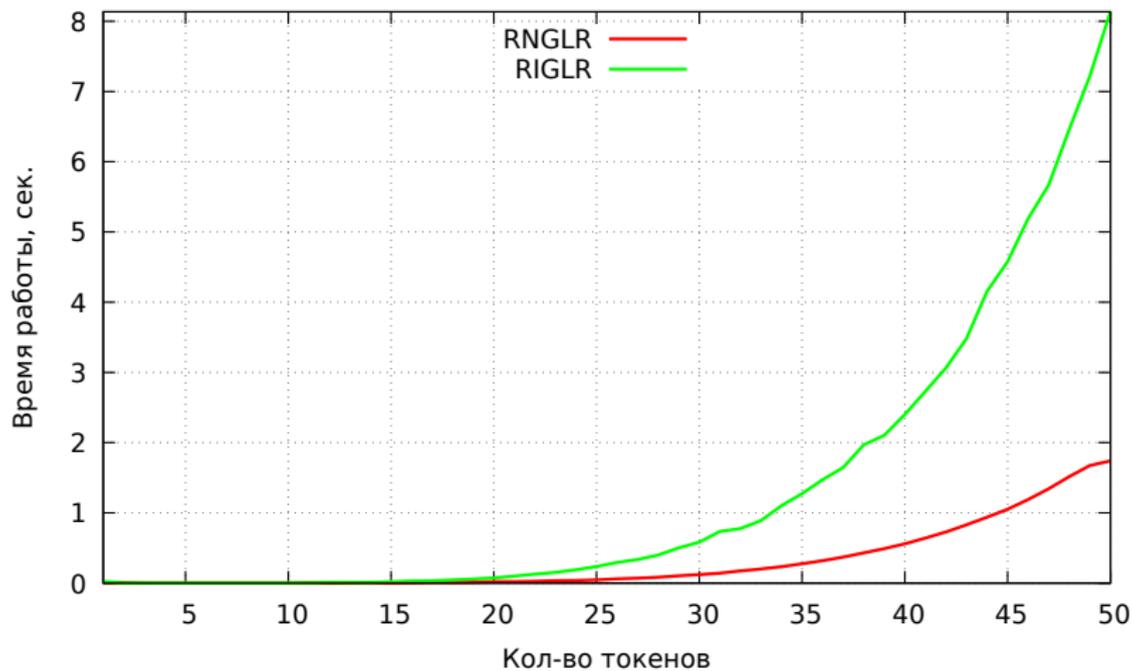
Токены	RNGLR	RIGLR
10	30	20
20	60	40
30	90	60
40	120	80
50	150	100

Кол-во вершин в стеке



Количество просмотров ребер стека

Эксперименты



Время работы

В ходе данной работы получены следующие результаты

- Изучен RIGLR-алгоритм синтаксического анализа
- Реализован генератор синтаксических анализаторов, основанный на RIGLR-алгоритме
- Проведено сравнение производительности анализаторов, созданных на основе RNGLR и RIGLR

Исходный код работы может быть найден в репозитории УС на GitHub (<https://github.com/YaccConstructor/YaccConstructor>). Автор работал под учетной записью *Lares77*.