



Библиотека GLL парсер-комбинаторов для .NET

Автор: Мелентьев Кирилл Игоревич, 3 курс
Научный руководитель: ст.пр. С.В. Григорьев

Санкт-Петербургский государственный университет
Кафедра системного программирования

18 мая 2016г.

- Синтаксический анализ
- Традиционный подход — генераторы синтаксических анализаторов
 - ▶ YACC, ANTLR, ...
- Парсер комбинаторы
 - ▶ Parsec — классическая библиотека для Haskell
 - ▶ FParsec — для F#
 - ▶ gll-combinators — GLL комбинаторы для Scala
- Распространенная проблема парсер-комбинаторов — отсутствие поддержки леворекурсивных правил в грамматике.

Обзор существующих решений

Feature\Lib	FParsec (F#)	XParsec (F#)	Attoparsec (Haskell)	gll-combinators (Scala)
Левая рекурсия	-	-	-	+
Вход абстрактного типа	-	+	-	-
Инкрементальный анализ	-	-	+	-

Постановка задачи

Целью работы является разработка библиотеки парсер-комбинаторов, поддерживающих произвольные КС грамматики для платформы .NET

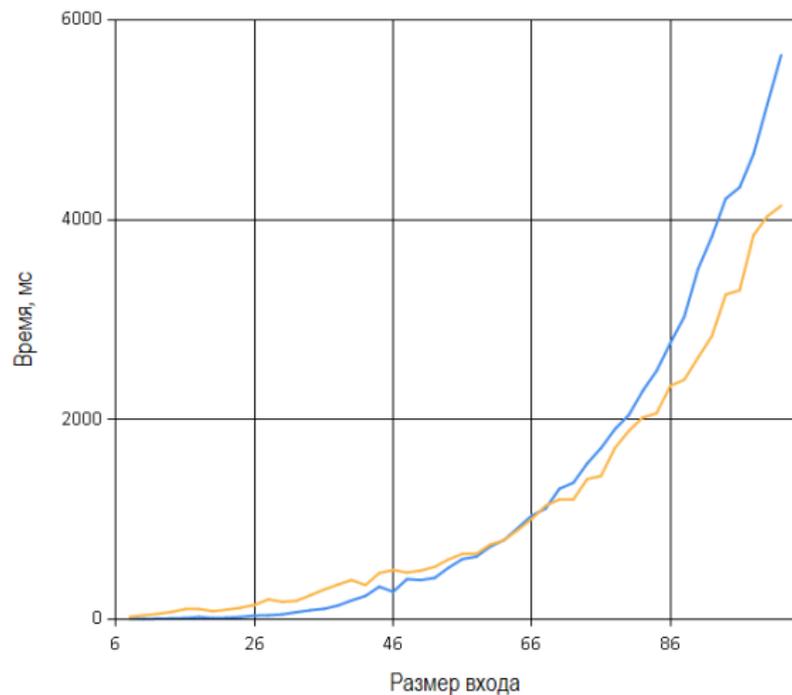
Задачи:

- Разработать библиотеку парсер-комбинаторов со следующими свойствами и возможностями:
 - ▶ Платформа реализации .NET
 - ▶ Поддержка произвольных КС грамматик
 - ▶ Поддержка абстрактного входного типа данных
 - ▶ Возможность инкрементального синтаксического анализа
- Провести сравнение производительности с существующими решениями

- Использован алгоритм GLL
 - ▶ Позволяет использовать произвольные грамматики
- Реализация комбинаторов для инкрементального синтаксического анализа:
 - ▶ На основе изменяемых структур — расходует дополнительную память
 - ▶ На основе неизменяемых структур — не копирует лишних данных, но производительность значительно ниже

Производительность

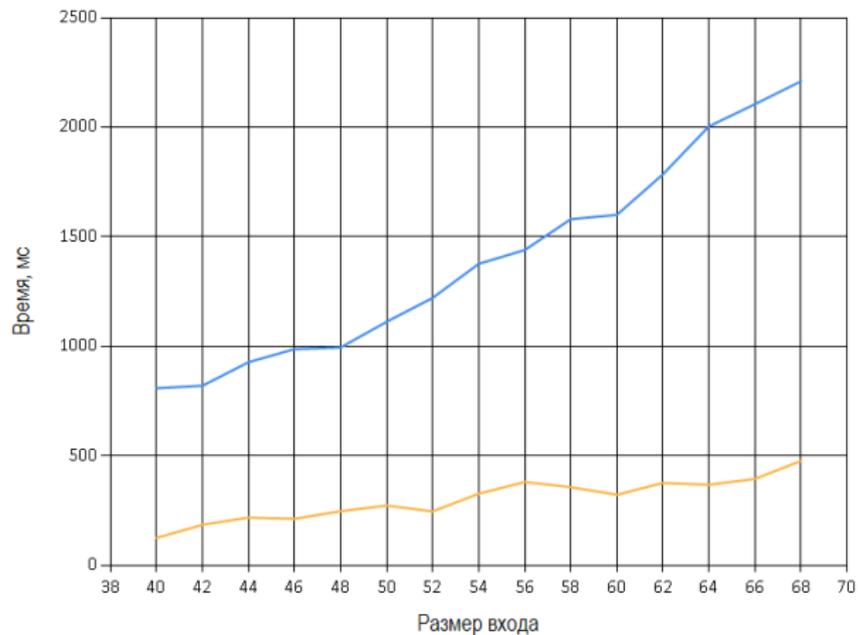
- $N ::= N N N \mid N N \mid '0'$
 - ▶ FsGll
 - ▶ gll-combinators (Scala)



Производительность

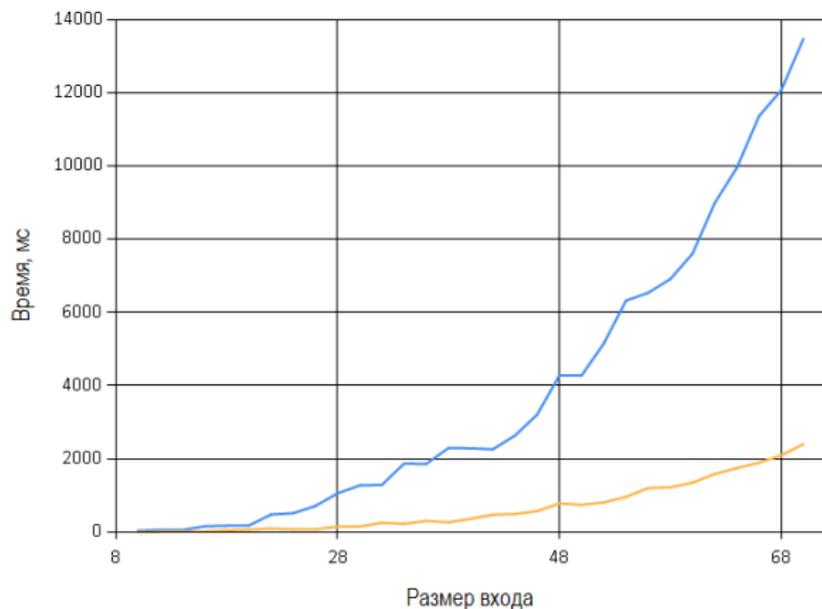
● ExtCalc

- ▶ FsGII
- ▶ FParsec



Производительность

- $N ::= N N N \mid N N \mid '0'$
 - ▶ FsGII, Incremental, Immutable
 - ▶ FsGII, Incremental, Mutable



- Реализована библиотека парсер-комбинаторов со следующими возможностями и свойствами:
 - ▶ Платформа реализации .NET
 - ▶ Поддержка произвольных КС грамматик
 - ▶ Поддержка абстрактного входного типа данных
 - ▶ Возможность инкрементального синтаксического анализа
- Проведено сравнение производительности с существующими решениями
- Выступление с докладом на конференции «Современные технологии в теории и практике программирования» в СПбПУ