Ослабленный синтаксический анализ динамически формируемых программ на основе алгоритма GLL

Автор: Рагозина Анастасия Константиновна, 661 группа
Научный руководитель: к.ф.-м.н., доцент Д.Ю. Булычев
Рецензент: инженер-программист ООО "ИнтеллиДжей Лабс" С.Д. Шкредов

Санкт-Петербургский государственный университет
Кафедра фундаментальной информатики и информационных технологий

27 мая 2016г.

Динамически формируемый код

- Динамический SQL

```
IF @X = @Y
    SET @TBL = ' #table1 '
ELSE
    SET @TBL = ' table2 '
SET @S = 'SELECT x FROM' + @TBL + 'WHERE ISNULL(n,0) > 1'
EXECUTE (@S)
```

- Встроенный SQL

```
SqlCommand myCommand = new SqlCommand(
    "SELECT * FROM table WHERE Column = @Param2",
    myConnection);
myCommand.Parameters.Add(myParam2);
```

Анализ метагеномных сборок

Задача: поиск маркерных генов в метагеномной сборке

- Метагеномная сборка представляется в виде конечного автомата, генерирующего различные геномы
  - Большие объёмы данных
- Структура маркерных генов задаётся грамматикой
  - Грамматика сильно неоднозначная

Существующие инструменты

- Подходы к анализу встроенных языков
  - Java String Analyzer, PHP String Analyzer, Alvor, IntelliLang, Varis
  - Абстрактный синтаксический анализ
  - Синтаксический анализ регулярных множеств на основе RNGLR
- Подходы к анализу метагеномных сборок
  - HMMER
  - REAGO, EMIRGE, Xander
  - Infernal

Постановка задачи

Целью работы является создание алгоритма синтаксического анализа регулярных множеств, применимого для анализа встроенных языков и для поиска в метагеномных сборках

Задачи:

- Разработать алгоритм синтаксического анализа регулярных множеств на основе алгоритма GLL
- Доказать корректность предложенного алгоритма
- Реализовать предложенный алгоритм в рамках проекта YaccConstructor
- Применить к задаче поиска на входных данных большого размера – метагеномных сборках
- Провести сравнение с алгоритмом на основе RNGLR

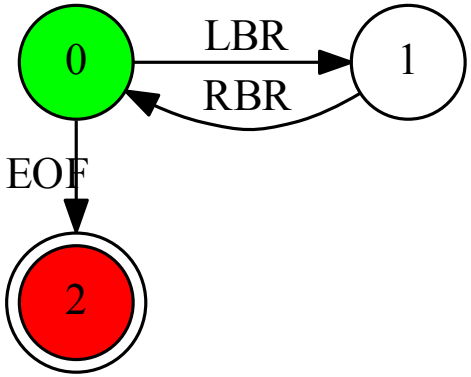
Алгоритм

- За основу взят алгоритм обобщённого анализа GLL
- **Вход**: эталонная КС-грамматика G и ДКА M без ϵ-переходов над алфавитом терминалов G
- **Выход**: лес разбора, содержащий все корректные деревья, построенные для цепочек, порождаемых входным КА

Пример работы алгоритма

```
string res = "";
for(i = 0; i < l; i++) {
    res = "()" + res;
}
```

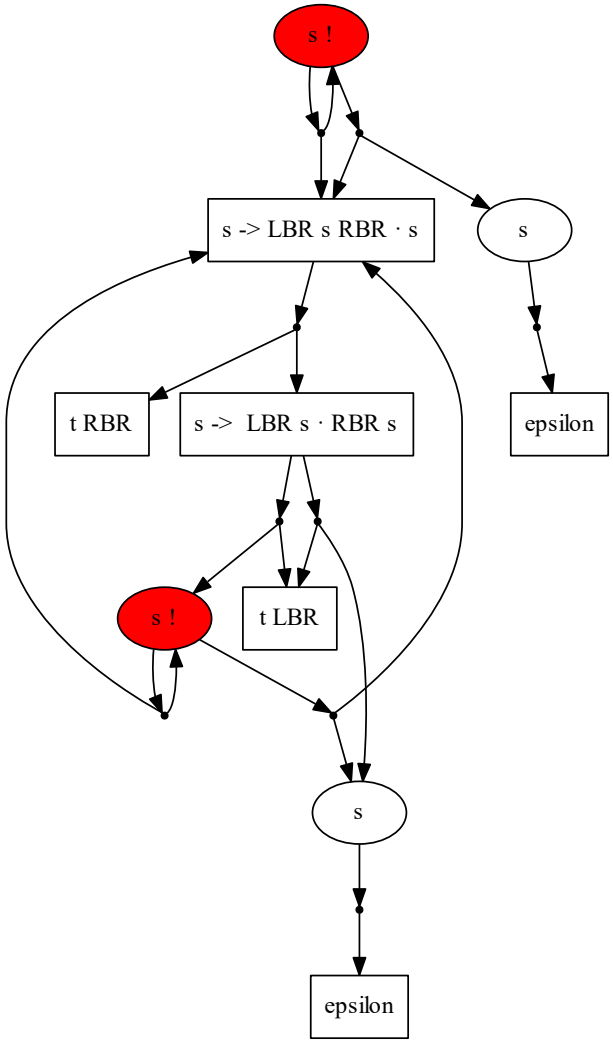
Аппроксимация:



Грамматика:

$$start ::= s$$
$$s ::= \texttt{LBR}\ s\ \texttt{RBR}\ s$$
$$s ::= \epsilon$$

Результат (SPPF):



Корректность алгоритма

Теорема (Завершаемость)

Алгоритм завершает свою работу для произвольного детерминированного конечного автомата и контекстно-свободной грамматики.

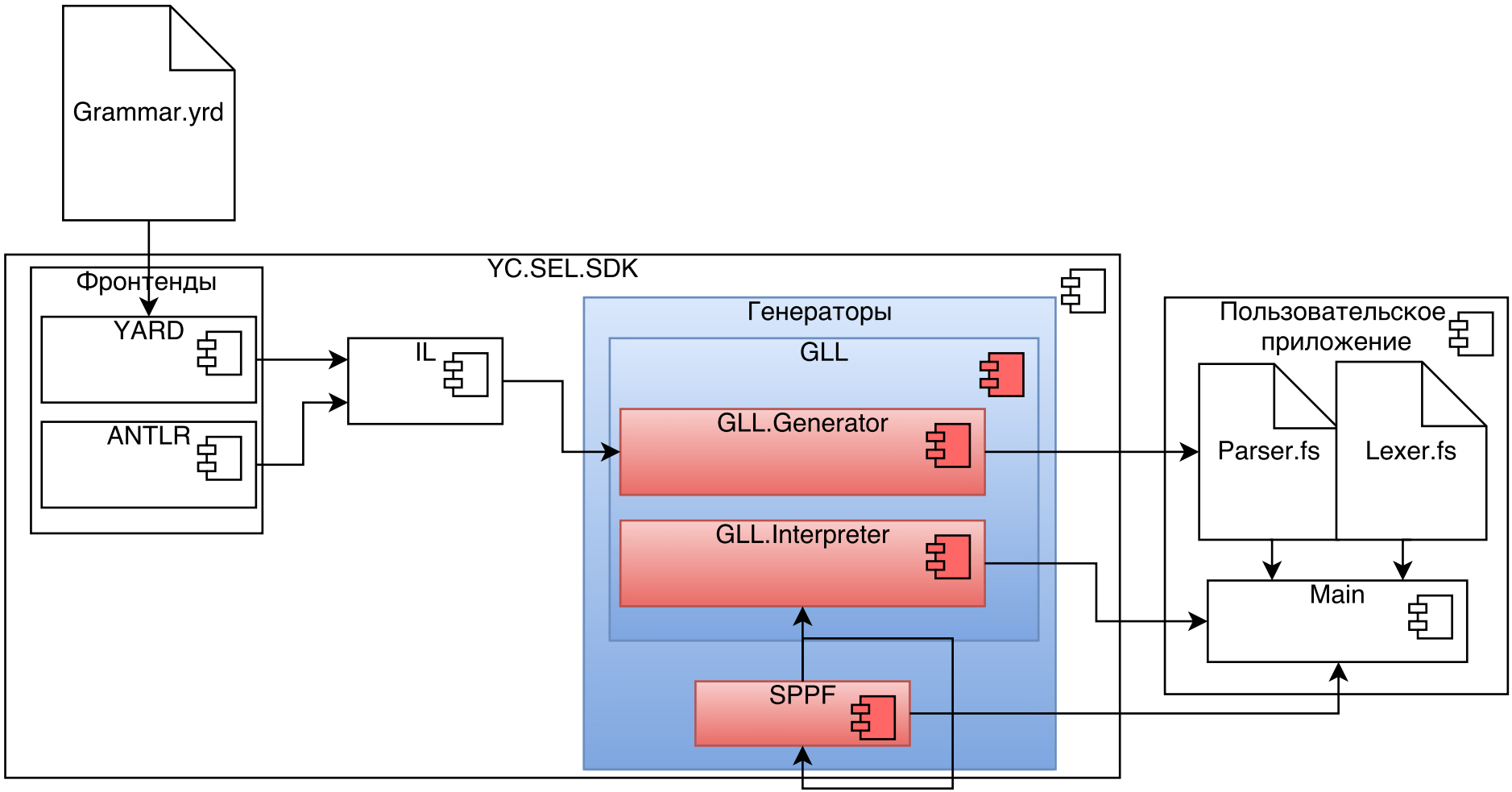
Теорема (Корректность)

Любое дерево, извлечённое из SPPF, является корректным.

Теорема (Полнота)

Пусть грамматика G порождает язык L. Тогда для каждого пути p в графе M, соответствующего строке ω из L, из SPPF может быть изъято корректное дерево.

Архитектура

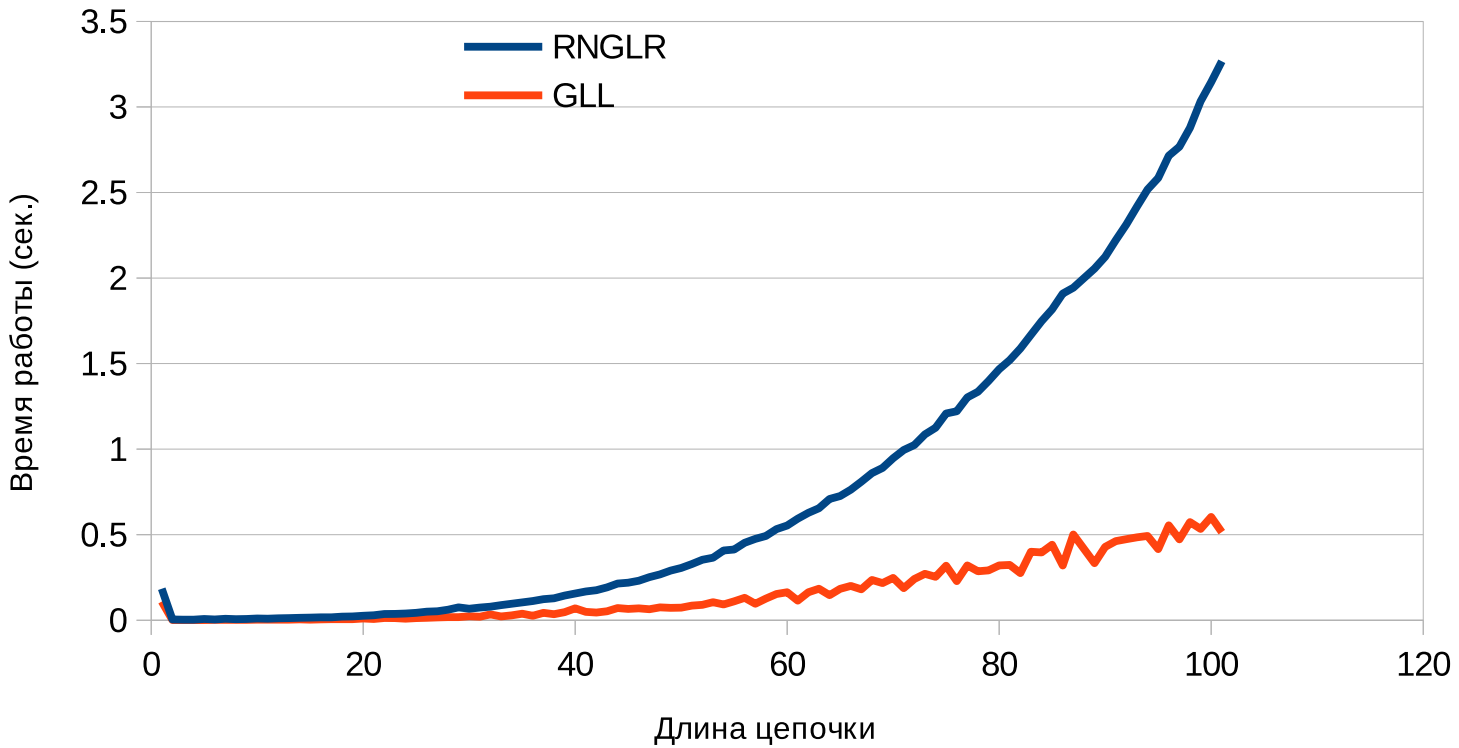


Реализация

- Алгоритм реализован в рамках проекта YaccConstructor, основным языком разработки является F#
- Для достижения высокой производительности потребовались специальные структуры данных
  - BlockResizeArray
    - ★ Лучшее использование памяти, чем у стандартного ResizeArray, на больших объёмах данных
    - ★ Была реализована ранее, дополнена в рамках работы
    - ★ Выложена в FSharpx.Collections, pull request принят, включена в бинарный пакет
  - Сжатые представления: несколько элементов сжаты в одно целое число
    - ★ Уменьшение объёма данных
    - ★ Ускорение поиска
- Модификация для обработки метагеномных сборок
  - Не строит SPPF
  - Более компактное представление входного КА

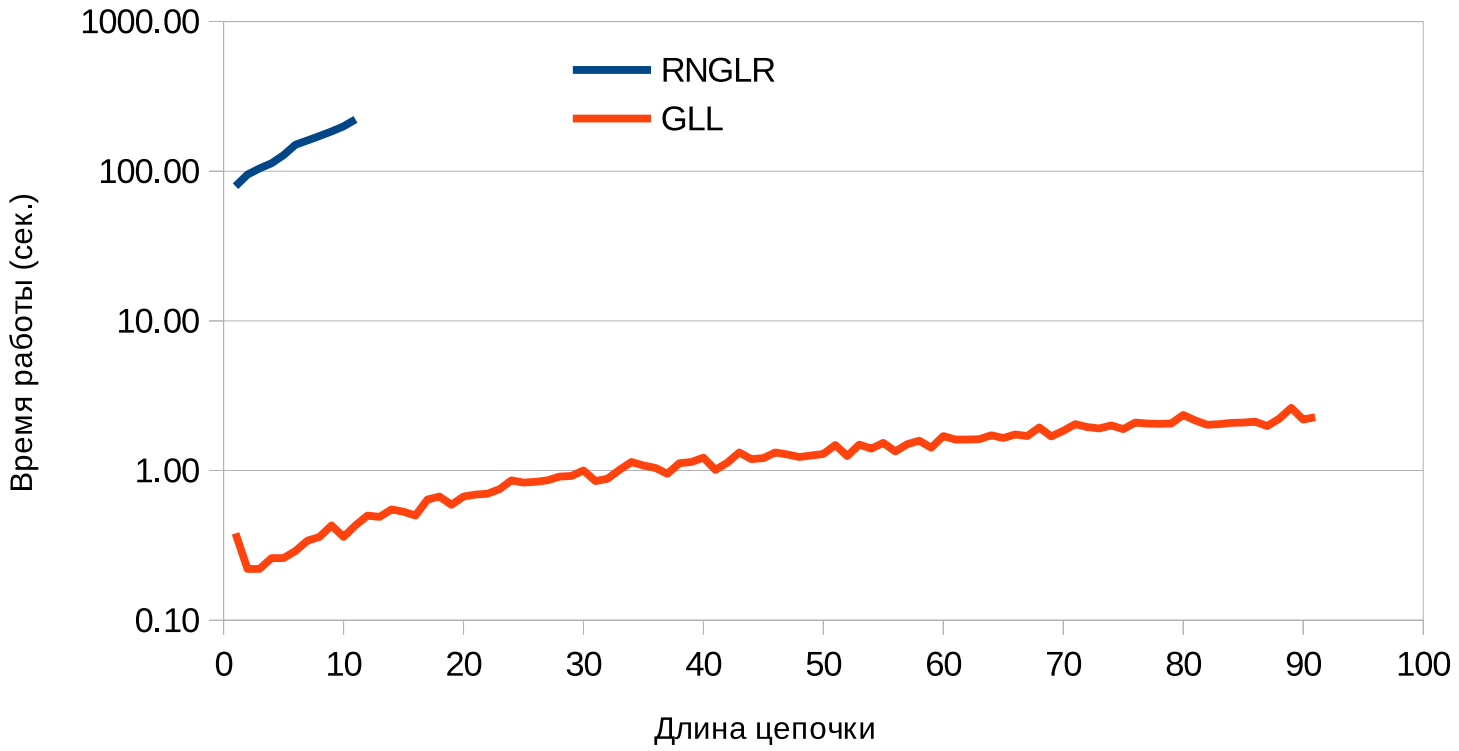
Эксперименты

- Произведены эксперименты по сравнению алгоритмов на основе алгоритма RNGLR и GLL



Эксперименты

- Произведены эксперименты по сравнению алгоритмов на основе алгоритма RNGLR и GLL для анализа данных большого объёма



Результаты работы

- Разработан алгоритм синтаксического анализа регулярных множеств на основе алгоритма GLL
- Доказана корректность предложенного алгоритма
- Предложенный алгоритм реализован в рамках проекта YaccConstructor
- Предложенный алгоритм применён к задаче поиска на входных данных большого размера
- Произведены эксперименты и сравнение
- Доклад “Обобщённый табличный LL-анализ” на конференции “ТМПА-2014”
- Публикация “Средство разработки инструментов статического анализа встроенных языков” в сборнике “Наука и инновации в технических университетах материалы Восьмого Всероссийского форума студентов, аспирантов и молодых ученых”
- Работа поддержана грантом УМНИК: договор №5609ГУ1/2014