Синтаксический анализ регулярных множеств

В рамках проекта лаборатории JetBrains

Автор: Вербицкая Екатерина Андреевна, 544 группа
Научный руководитель: ст.пр. С.В. Григорьев
Рецензент: программист "ИнтеллиДжей Лабс" А.А. Бреслав

Санкт-Петербургский государственный университет
Кафедра системного программирования

18 июня 2015г.

Встроенные выражения

- 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);
```

Схема анализа встроенного кода

Фрагмент кода

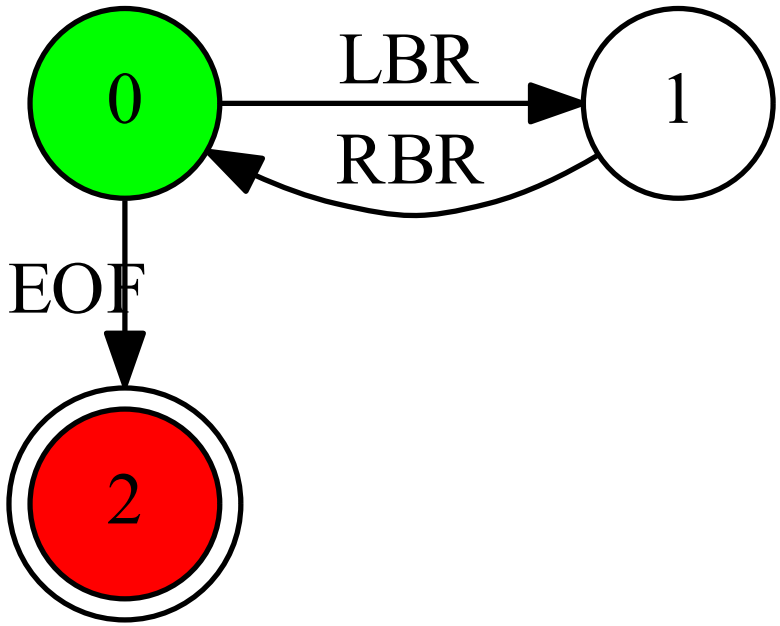
```
string res = "";
for(i = 0; i < l; i++) {
    res = "()" + res;
}
```

Регулярная аппроксимация

```
("()")*
```

Множество значений

```
{"",
 "()",
 "()()",
 ...
 "()"^l,
}
```



Существующие инструменты

- Java String Analyzer, Alvor
  - Регулярная аппроксимация
- PHP String Analyzer
  - КС-аппроксимация
- Kyung-Goo Doh et al.
  - Решение уравнений потока данных над множеством LR-стеков
- Недостатки
  - Плохо расширяемы
  - Не строят структурного представления кода

Постановка задачи

Целью работы является разработка алгоритма, применимого для синтаксического анализа встроенных языков

Задачи:

- Разработать алгоритм синтаксического анализа регулярной аппроксимации динамически формируемых строковых выражений, строящий конечное представление леса разбора
- Доказать корректность алгоритма
- Реализовать предложенный алгоритм
- Провести апробацию

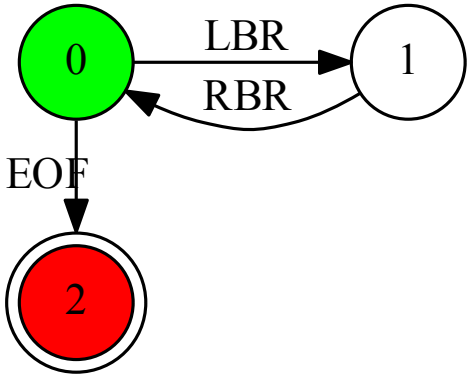
Алгоритм

- **Вход**: эталонная ДКС-грамматика G и граф ДКА без ϵ-переходов над алфавитом терминалов G
- **Выход**: конечное представление множества деревьев, соответствующих всем корректным цепочкам, принимаемым входным автоматом

Алгоритм

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

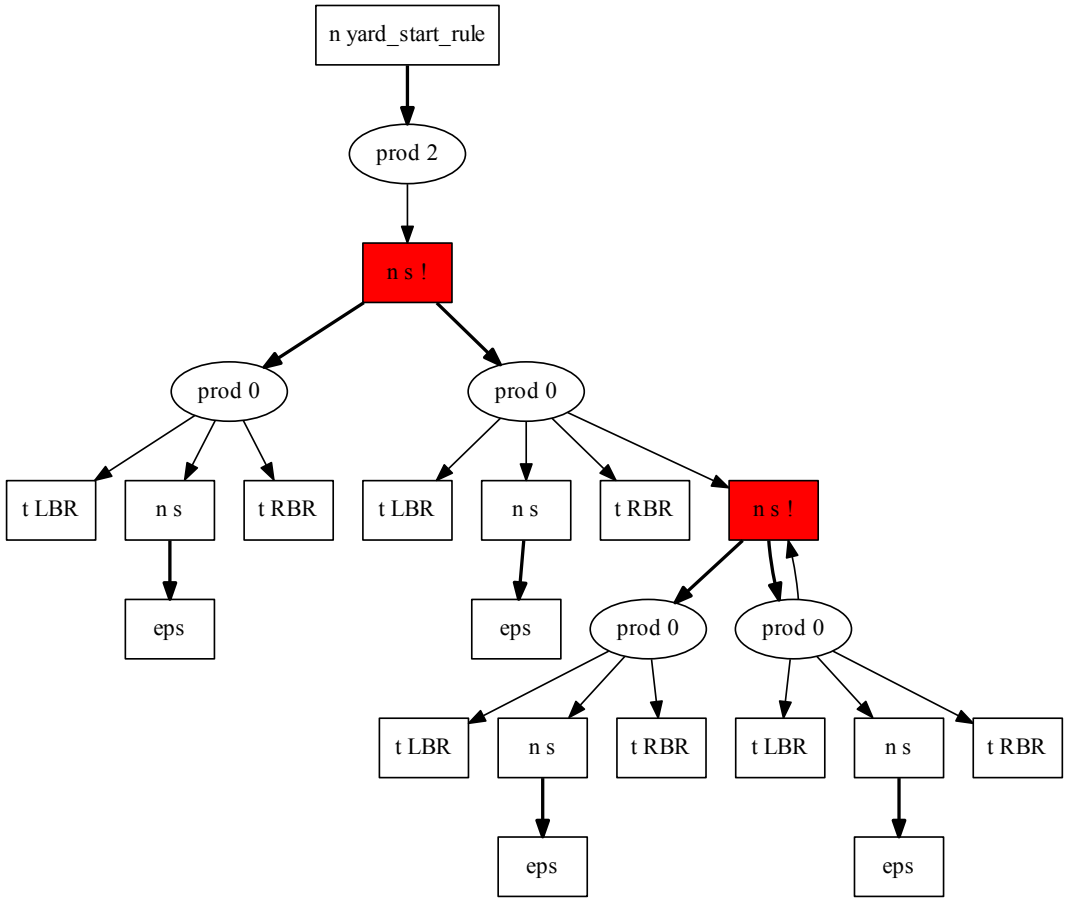
Аппроксимация:



Грамматика:

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

Результат (SPPF):

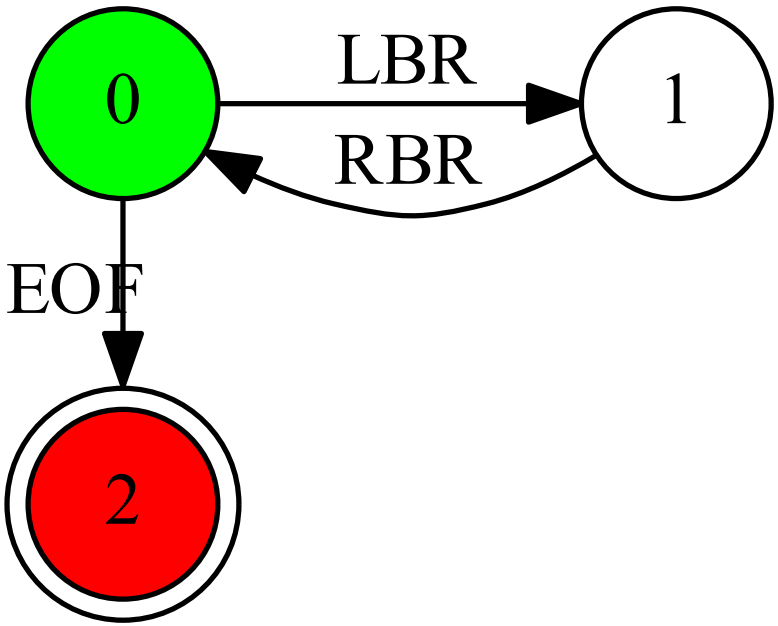


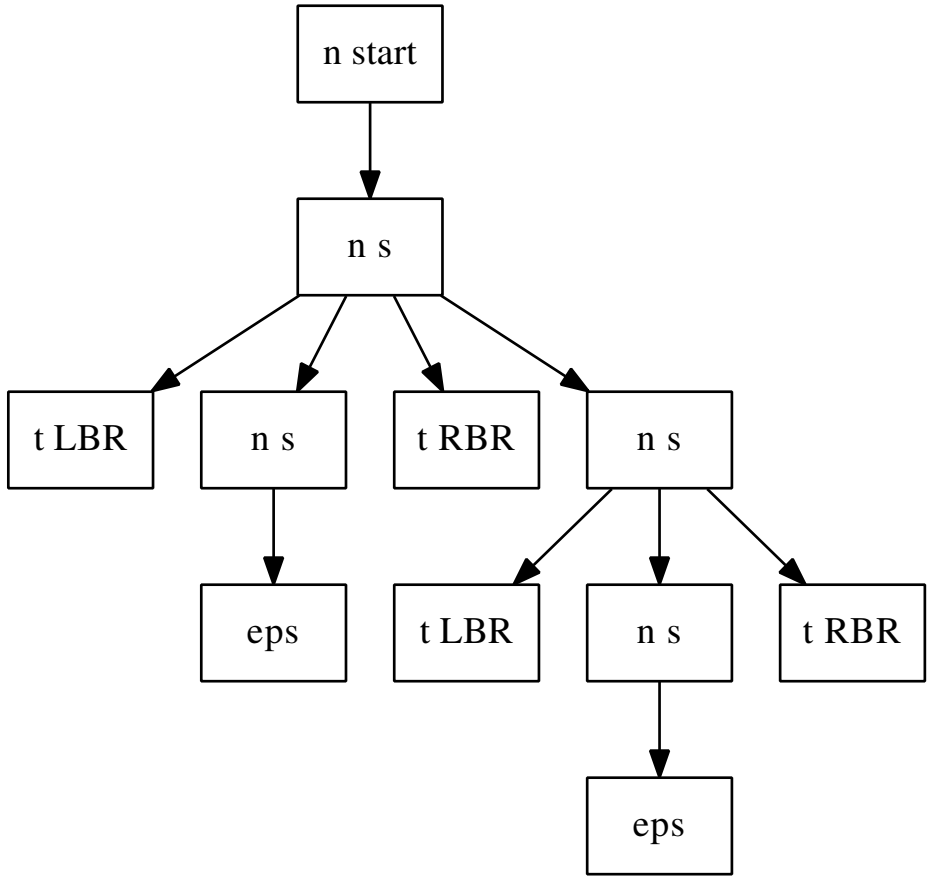
Алгоритм

- Управление стеком и построение леса разбора (SPPF) осуществляется, как в RNGLR-алгоритме
- С каждой вершиной входного графа ассоциируется множество LR-состояний синтаксического анализатора
- Последовательное построение стека GSS во время обхода входного графа
- Для управления порядком обработки вершин входного графа используется очередь. Вершина добавляется в очередь, когда в GSS добавляется новое ребро с концом в этой вершине

Алгоритм: корректность

Корректное дерево – дерево вывода цепочки, накопленной вдоль некоторого пути в графе





Алгоритм: корректность

Теорема (Завершаемость)

Алгоритм завершается для любой ДКС-грамматики G и любого ДКА без $\epsilon-$переходов.

Теорема (Корректность)

Любое дерево, извлечённое из SPPF, корректно.

Теорема (Корректность)

Для строки, сооответствующей любому пути p во входном графе, имеющей вывод в эталонной грамматике G, корректное дерево, соответствующее p, может быть извлечено из SPPF.

Реализация

- Алгоритм реализован как часть проекта YaccConstructor на языке F#
- Переиспользован генератор RNGLR-таблиц и структуры данных GSS и SPPF
  - Дипломная работа выпускника кафедры системного программирования Авдюхина Дмитрия

Апробация

- Данные из проекта по миграции ИС с MS-SQL на Oracle Server
- 2,7 миллиона строк кода, 2430 запросов, 2188 из них разобрано
- Количество запросов, которые не удалось разобрать из-за таймаута, сократилось с 45 до 1

Результаты

- Разработан алгоритм синтаксического анализа регулярной аппроксимации динамически формируемых строковых выражений, строящий конечное представление леса разбора
- Доказана завершаемость и корректность алгоритма
- Выполнена реализация алгоритма на языке программирования F# в рамках исследовательского проекта YaccConstructor
- Проведена апробация

- Подана статья “Relaxed Parsing of Regular Approximations of String-Embedded Languages” на конференцию PSI-2015