

Диагностика ошибок при анализе встроенных языков

**В рамках проекта лаборатории
JetBrains**

**Автор: студент группы №444, Вербицкая Екатерина
Научный руководитель: Григорьев Семён**

Встроенные языки

- Динамический SQL:

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

- Встроенный SQL:

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

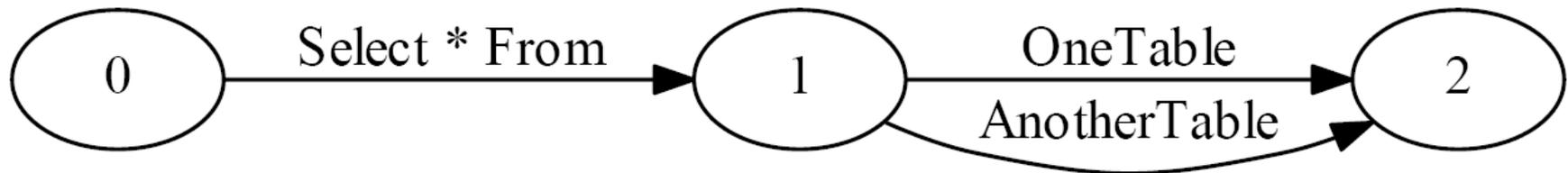
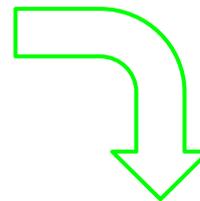
AbstractYaccConstructor

- Платформа для создания инструментов для работы со встроенными языками
 - Плагин к ReSharper
 - Плагин должен сообщать пользователю об ошибках в коде, подобно стандартным IDE
- Независима от обрабатываемых языков, потому нужны общие механизмы

Абстрактный анализ

- Вход структурирован в виде графа
- В основе — **GLR-алгоритм** анализа

{ “Select * From OneTable”,
“Select * From AnotherTable” }

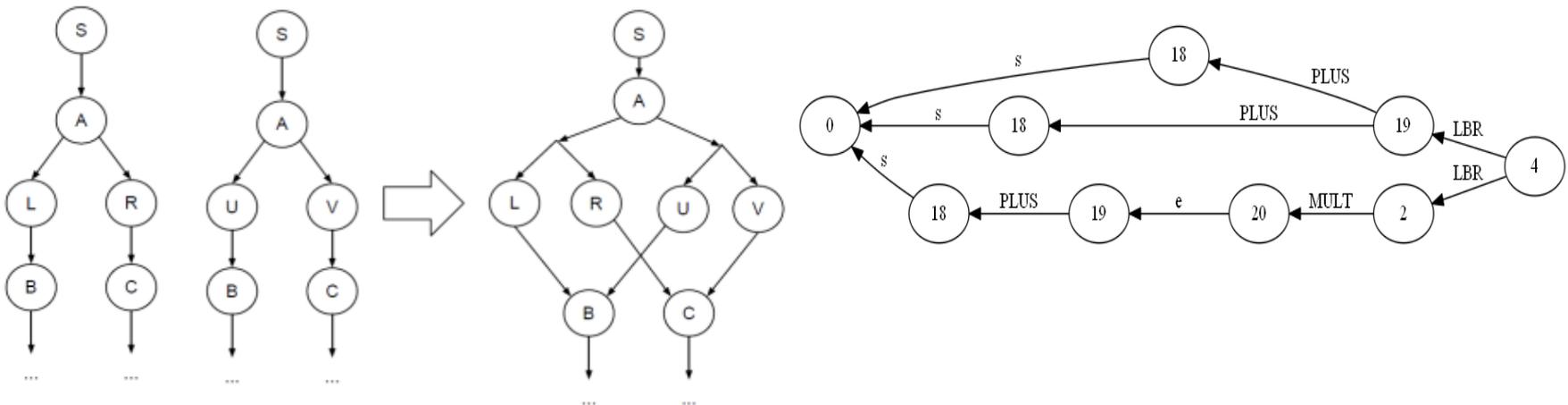


Постановка задачи

- Изучить механизм работы абстрактного синтаксического анализа, основанного на GLR-алгоритме
- Исследовать особенности обнаружения ошибок при абстрактном анализе
- Реализовать на основе результатов исследования механизм обнаружения ошибок во встроенных языках

GLR-алгоритм

- Обобщенный восходящий анализатор
- Для неоднозначных грамматик
- 2 вида конфликтов: Shift/Reduce и Reduce/Reduce
- Структурированный в виде графа стек (Graph Structured Stack)
- Сжатое представление леса разбора (Shared Packed Parse Forest)

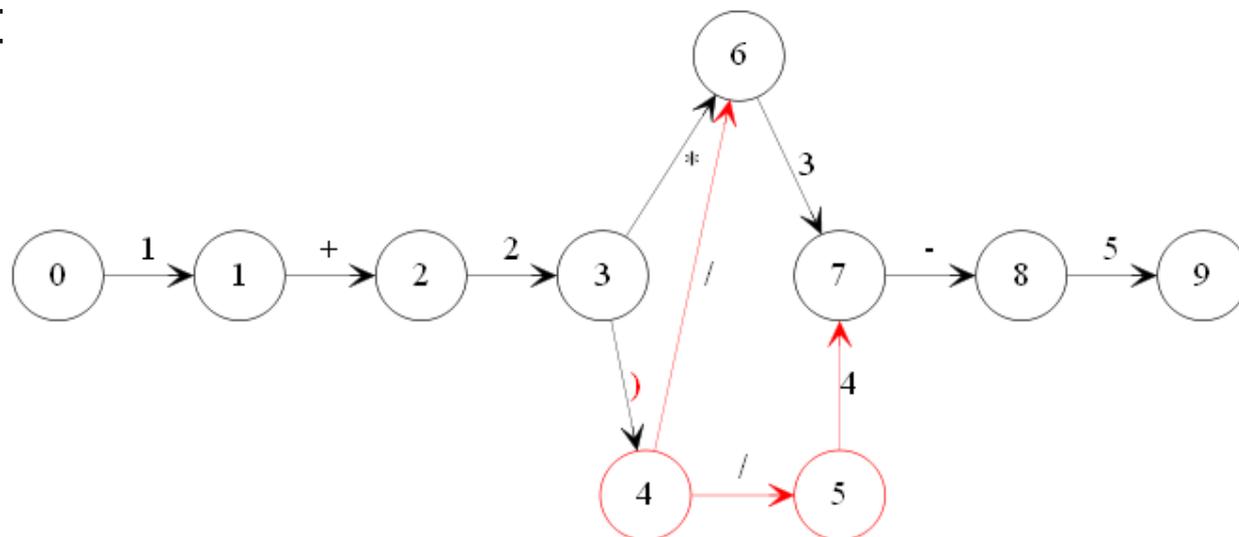


Абстрактный анализ: отличие от классического GLR

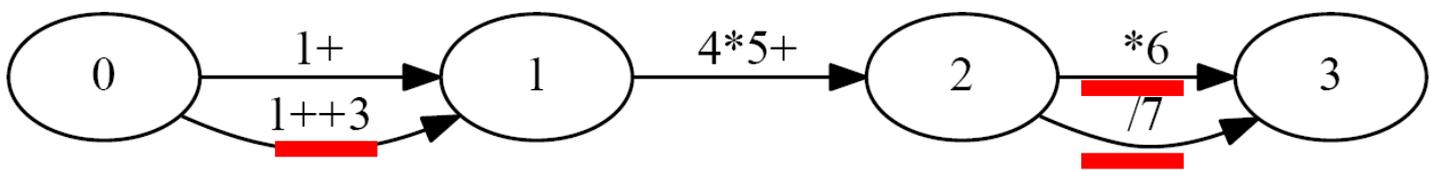
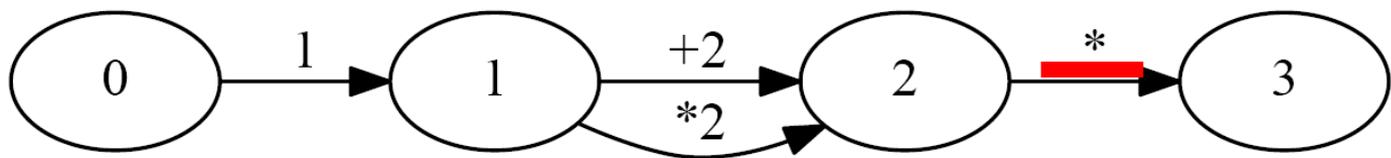
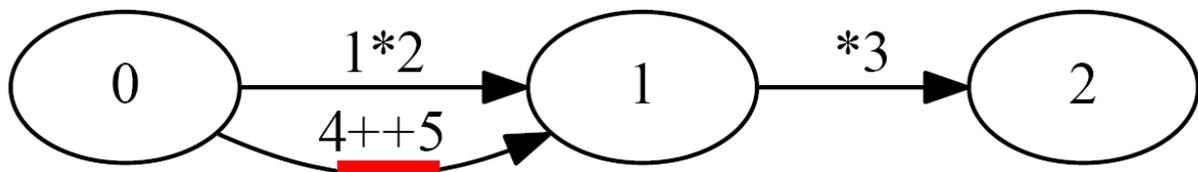
- Новый вид конфликтов Shift/Shift: соответствуют **разветвлениям** во входном графе
- Ветки в графе соответствуют **веткам в стеке**
- Обработка ошибок **наследуется** от оригинального алгоритма

Проблемы

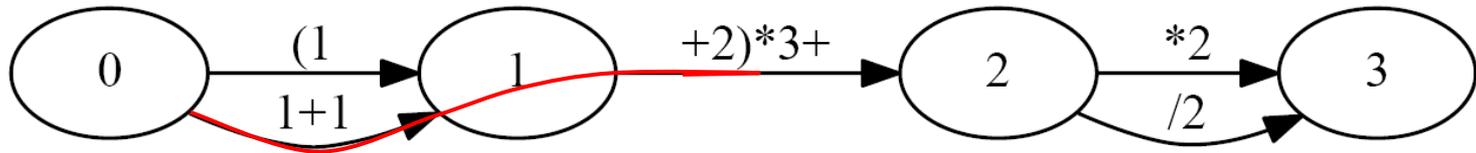
- Нужно возвращать лес разбора для корректных выражений и список ошибок для некорректных
- Для обычного GLR умершая ветка — нормально, для абстрактного не всегда
- Пропускать токены в графе сложнее, чем в линейном пс



Можем обработать



Не могут быть обработаны из-за специфики GLR



- Не знаем, откуда взялось плохое состояние во время обработки “)”.
- GLR уничтожает такие ветки, считая, что имела место ошибка выбора варианта при конфликте
- Можно протаскивать информацию, что имел место именно Shift/Shift “конфликт”, но это экспонента (количество путей от начала до вершины — степень ветвления)

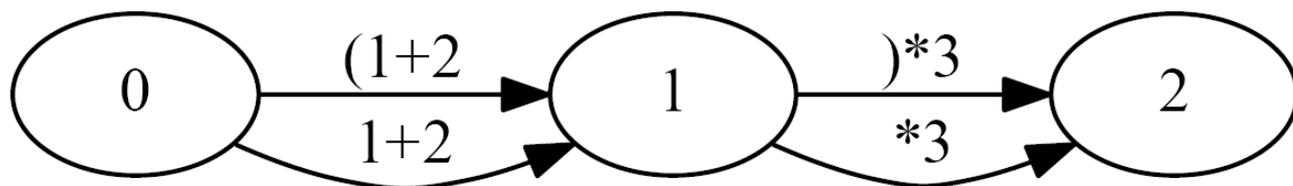
Недостаток аппроксимации: непорождаемые выражения

- В графе могут быть пути, **принципиально** не порождаемые и некорректные в реальном мире
- Данную проблему нужно решать во время семантического анализа

let t = if cond then "(1+2" else "1+2"

let h = if cond then ")*3" else "*3"

eval (t+h)



Результаты

- Изучен механизм работы абстрактного синтаксического анализа, основанного на GLR-алгоритме
- Выявлен набор случаев, при которых обнаружение ошибок работает некорректно
- Некоторые из проблем удалось решить
- Некоторые из проблем решить не удастся из-за специфики работы GLR-алгоритма
- Тезисы работы изложены в статье, принятой на Workshop on Science Intensive Applied Software