

Абстрактный синтаксический анализ на основе GLR-алгоритма

Автор: Григорьев Семён Вячеславович

Лаборатория JetBrains на Математико-Механическом факультете
Санкт-Петербургского государственного университета

24 октября 2013г.

- Динамический SQL

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

- JavaScript в Java

```
String script =
    "function hello(name) print('Hello, ' + name); ";
engine.eval(script);
Invocable inv = (Invocable) engine;
inv.invokeFunction("hello", "Scripting!!!" );
```

- Ошибки в динамически формируемых выражениях обнаруживаются лишь во время выполнения.
- Динамически формируемые выражения – код на некотором языке. Его нужно соответствующим образом поддерживать и обрабатывать.
 - ▶ Поддержка в IDE.
 - ▶ Реинжиниринг ПО, разработанного с использованием встроенных языков.

- Статический анализ для встроенных языков. Абстрактный лексический и синтаксический анализы.
 - ▶ Поддержка в IDE:
 - ★ Многие ошибки можно искать без запуска программы.
 - ★ Автодополнение, рефакторинг.
 - ★ Возможны трансформации.
 - ▶ Реинжиниринг:
 - ★ Статический анализ.
 - ★ Трансформация (трансляция).

Пример работы

```
public void Select(int cond)
{
    var baseQuery = "select x, y + 1, z " ;
    string fields;
    switch (cond)
    {
        case 1:
            fields = "alias1";
            break;
        case 2:
            fields = "alias1";
            break;
        default:
            fields = "fld3 alias1";
            break;
    }
    var tableName = "defaultTable";
    if (cond == 1) tableName = "table2";
    Program.ExecuteImmediate(baseQuery + fields + "from" + tableName);
}
```

Syntax error. Unexpected token "IDENT"("alias1fromdefaultTable")

Syntax error. Unexpected token "IDENT"("alias1fromtable2")

Существующие решения

- Alvor – плагин для Eclipse для статической проверки встроеного в Java SQL.
- Java String Analyzer – статический анализатор динамических выражений для Java.
- PHP String Analyzer – статический анализатор динамических выражений для PHP.

В основе этих реализаций лежит (G)LR-анализ и они используют дополнительные структуры данных для организации стека и вычисления семантики.

Как это работает

- Для каждого выражения строится конструкция, аппроксимирующая множество его возможных значений.
 - ▶ Data-flow уравнение
 - ▶ **Граф**
 - ▶ Регулярное выражение
- Выполнение лексического, синтаксического анализа над графом – абстрактный анализ.

Пример

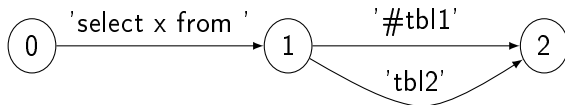
- Код:

```
var tbl1 = "#tbl1"
```

```
var tbl2 = "tbl2"
```

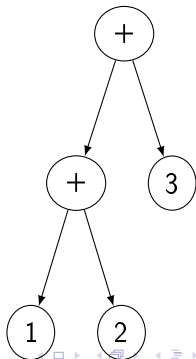
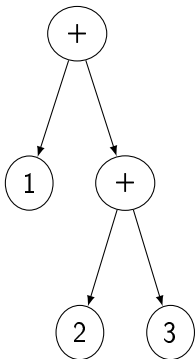
```
execute ("select x from " + if cond then tbl1 else tbl2)
```

- Входной “поток” для анализатора:



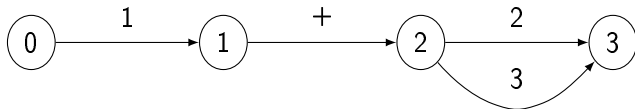
GLR

- Структурированный в виде графа стек.
- Ветвления при возникновении конфликтов (Reduce/Reduce, Shift/Reduce)
- Пример
 - ▶ Грамматика:
e: e '+' e | NUM
 - ▶ Вход: 1 + 2 + 3

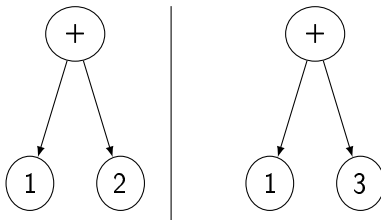


Абстрактный GLR

- Добавим Shift/Shift конфликты — конфликты, возникающие при ветвлении входного потока.
- Вход:



- Результат:



- Алгоритм абстрактного синтаксического анализа на основе GLR-алгоритма.
 - ▶ Используется обычный структурированный в виде графа стек для GLR.
 - ▶ Для представления результатов анализа (леса разбора) используется компактное внутреннее представление.
 - ▶ Генератор анализаторов.
- Плагин для ReSharper.
 - ▶ Расширяемая архитектура, позволяющая легко поддержать любой встроенный язык. Внешний язык должен поддерживаться в ReSharper.

- Поддержка встроенных языков в IDE:
 - ▶ Интерактивная ("на лету")
 - ▶ "Офлайновая" проверка (ручной запуск)
- Автоматизированный реинжиниринг ПО, разработанного с применением встроенных языков.

Информация о проекте

- Контакты:
 - ▶ Григорьев Семён Вячеславович: Semen.Grigorev@jetbrains.com
- YaccConstructor: <http://recursive-ascent.googlecode.com>
- Plug-in for ReSharper
 - ▶ Wiki: https://code.google.com/p/recursive-ascent/wiki/ReSharper_AbstractAnalysis
 - ▶ NuGet package: <https://www.myget.org/gallery/yc>