Abstract Parsers Generator Based on a Modified RNGLR Algorithm

Author: Semyon Grigorev

JetBrains

2013г.

Introduction

- String-embedded languages:
  - Embedded SQL (in C#, Java, C++)
  - Dynamic SQL
  - HTML pages generation
  - Other textual DSLs
- Static analysis of embedded statements.
  - Syntax errors checking and other analysis:
  - IDE support
    - ★ Syntax highlighting
    - ★ Refactorings
    - ★ Errors notification

String-embedded languages: examples

- Dynamic SQL

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

- JavaScript in Java

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

Abstract parsing

Main points of "classical" abstract parsing

- Based on (G)LR algorithm
  - Tables generator could be reused.
- Additional data structures
  - "Abstract" stack
  - Parsing forest

There is a set of tools based on it.

- Alvor: Eclipse plug-in for SQL ebedded into Java checking.
- Java String Analyzer: tool for analyzing the flow of strings and string operations in Java programs.
- PHP String Analyzer: static program analyzer that approximates the string output of a PHP program.

GLR parsing

- For ambiguous CFG processing.
- Shift\Reduce and Reduce\Reduce conflicts.
- Graph Structured Stack – GSS.
  - Branches on conflicts.
  - States merging.

Our approach

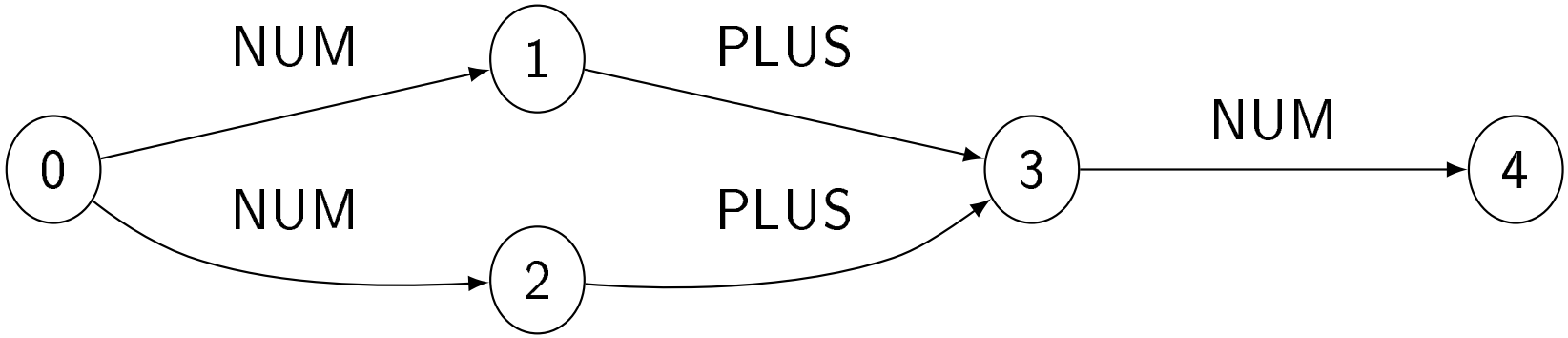
- We work with RNGLR – Right Nulled GLR.
- Shift\Shift conflicts — branches in input "stream". Impossible for sequential input.
  - RNGLR should be extended for process this conflicts.
- Reusing of inner structures: GSS and parsing result (graph-structured also).
  - Shift\Shift conflicts — new branches in GSS.
  - Parsing result is a graph which contains all possible trees.

Example: input

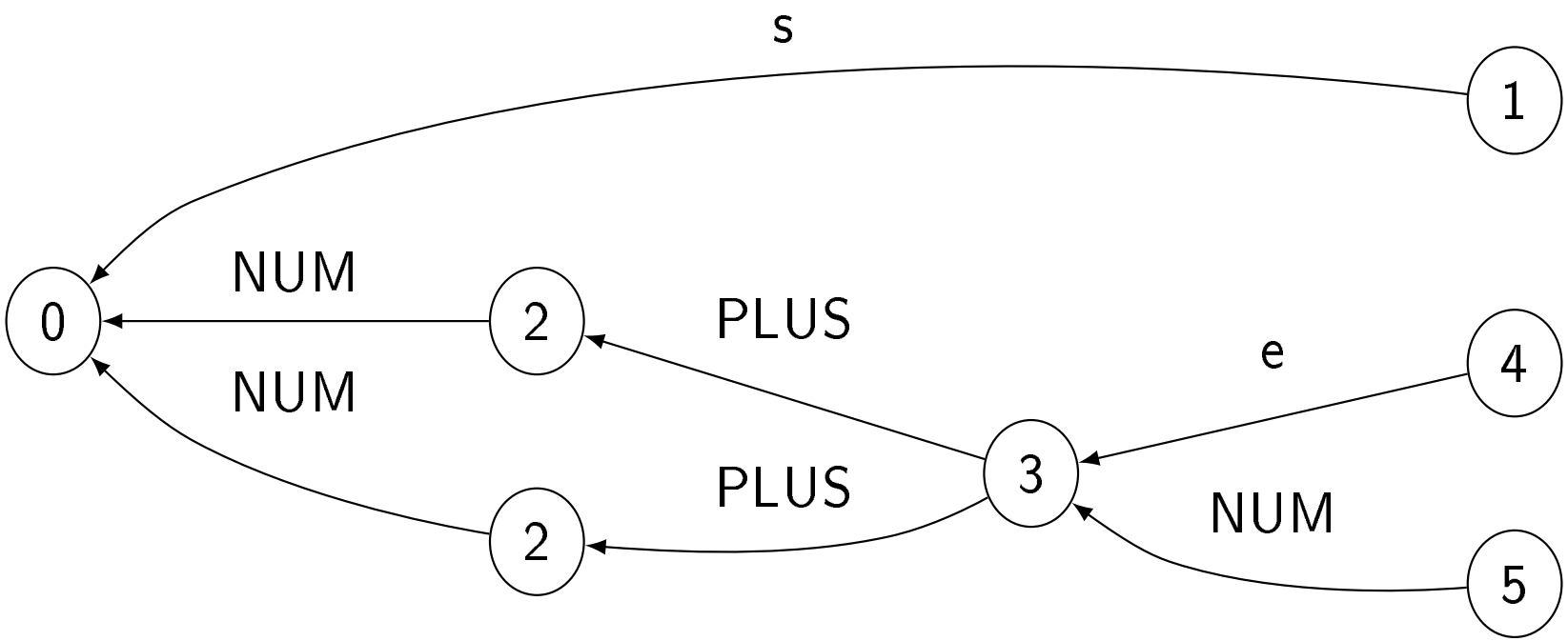
- Grammar:

```
s: NUM PLUS e
e: NUM
```

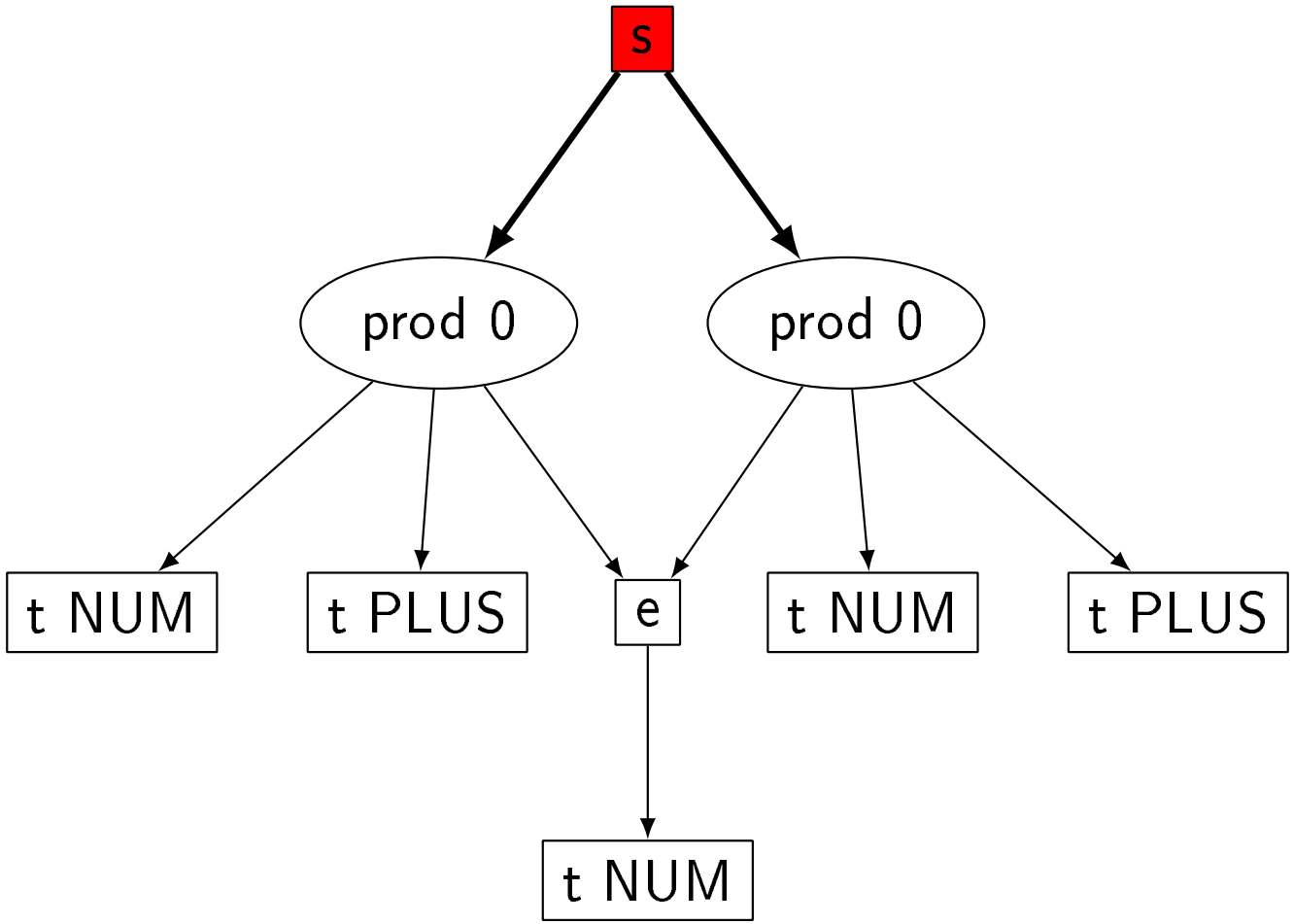
- Input graph:



Example: stack



Example: parsing result



Results

- Abstract parsing algorithm based on modified RNGLR is implemented.
  - GSS is reused.
  - Compact inner representation of parsing results with node reusing is used.
  - Parsers geneartor is implemented.
- Plug-in for ReSharper in active development.
  - Modular architecture is allow to create extensions for any string-embedded language.

Demo

```
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")

Contacts

- Semyon Grigorev:
  - Semen.Grigorev@jetbrains.com
  - rsdpisuy@gmail.com

- YaccConstructor: http://recursive-ascent.googlecode.com