From abstract parsing to abstract translation

Research project

Author: Grigorev Semyon

Saint-Petersburg State University
The faculty of Mathematics and Mechanics

29.05.2014

String-embedded languages

- 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!!!" );
```

Problems

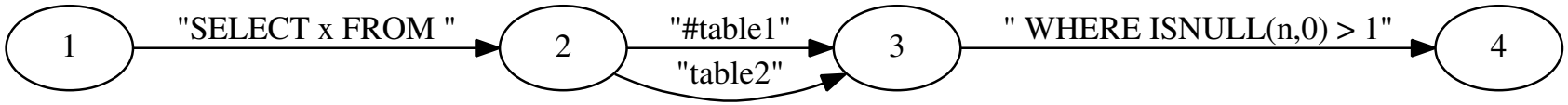
- Strings are expressions in programming language
  - They can contain errors
  - It may be necessary to transform them
  - Any other problems of programming languages may occure

Related work

- Kyung-Goo Doh, Hyunha Kim, David A. Schmidt
  - Combination of LR-based parsing algorithm and data-flow analysis to process string-embedded languages
    - ★ We try to parse an approximation of set of dymaically constructed expression: data-flow equation, **graph**, etc
  - We can use attributed grammars to specify semantics actions
  - Naive implementation of proposed algorithm has performance and space issues
- Alvor, Java String Analyzer, PHP String Analyzer are not usable for transformations

Approximation

- ```
  IF @X = @Y
      SET @TABLE = '#table1'
  ELSE
      SET @TABLE = 'table2'
  EXECUTE
      ('SELECT x FROM ' + @TABLE + ' WHERE ISNULL(n,0) > 1')
  ```
- Set of values:
  {'SELECT x FROM #table1 WHERE ISNULL(n,0) > 1' ;
  'SELECT x FROM table2 WHERE ISNULL(n,0) > 1'}
- Approximation:


  ```

Real world example

DBMS migration from MS-SQL (T-SQL) to Oracle server (PL-SQL)

- $>$ 2 mln lines of code
- 3000 hotspots (EXECUTE(string) statements)
  - More than 50% of them can have more than one value
  - 212 is a maximum number of expression-generating operators for one expression
  - 40 is average number of expression-generating operators

Real world example

DBMS migration from MS-SQL (T-SQL) to Oracle server (PL-SQL)

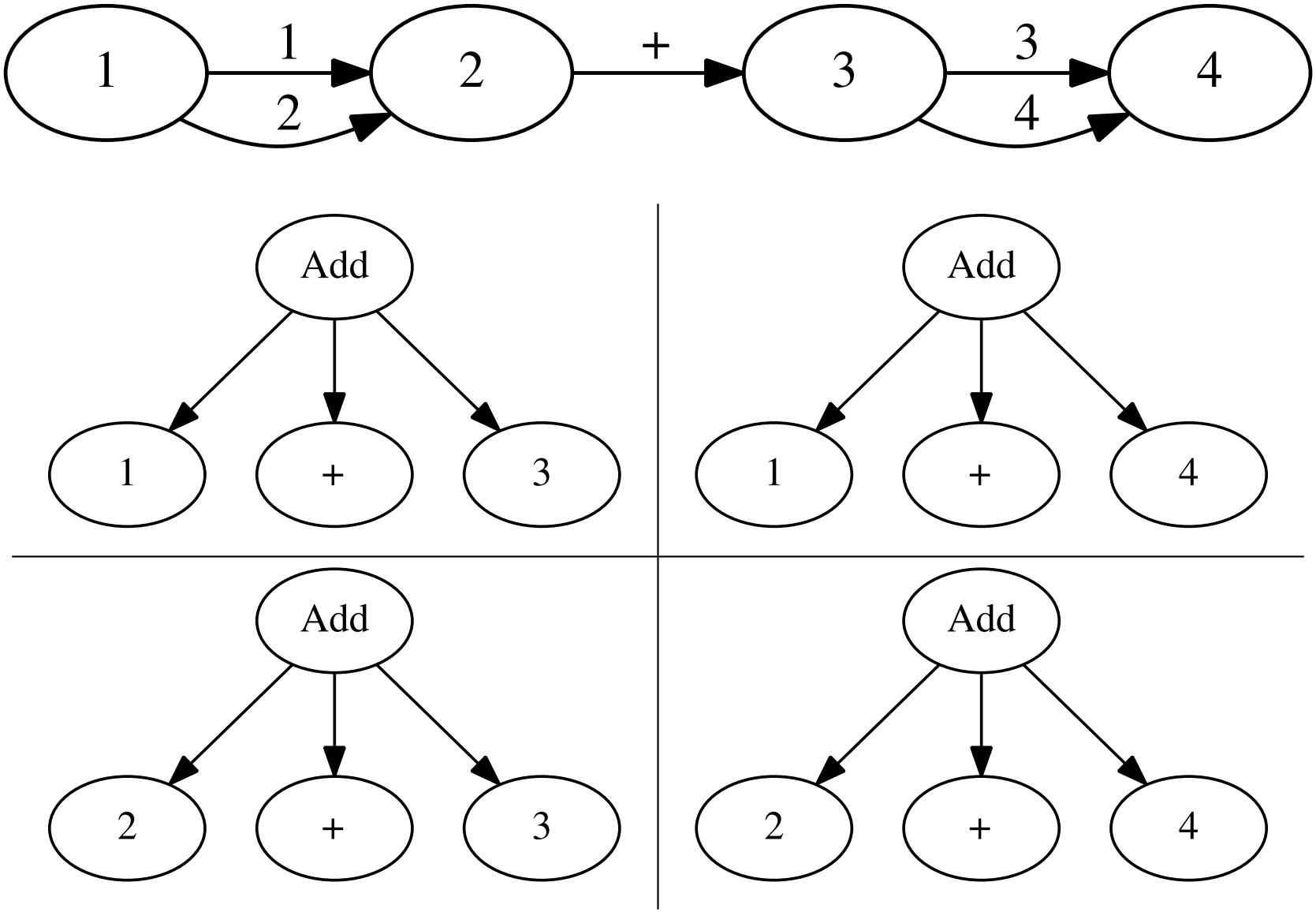
- $>$ 2 mln lines of code
- 3000 hotspots (EXECUTE(string) statements)
  - More than 50% of them can have more than one value
  - 212 is a maximum number of expression-generating operators for one expression
  - 40 is average number of expression-generating operators
- $>$ 16 Gb RAM in use and not finished in 5 hours because we get a huge number of trees

Solution

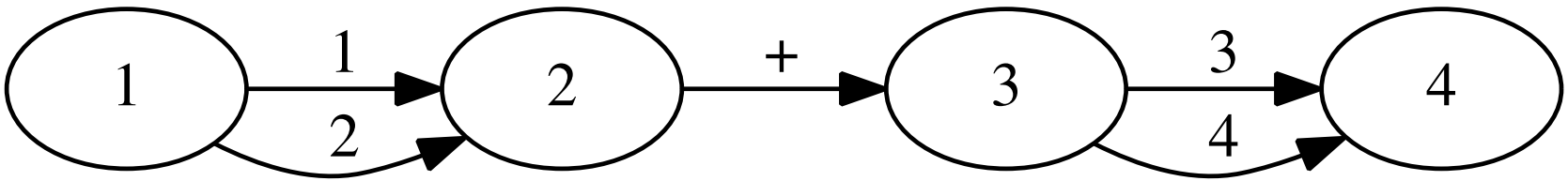
Run time parsing results filtration

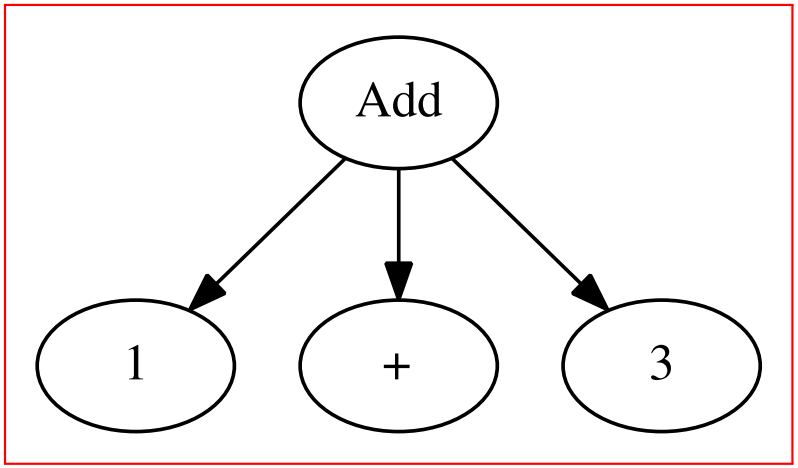
- Stacks filtration
- Forest filtration

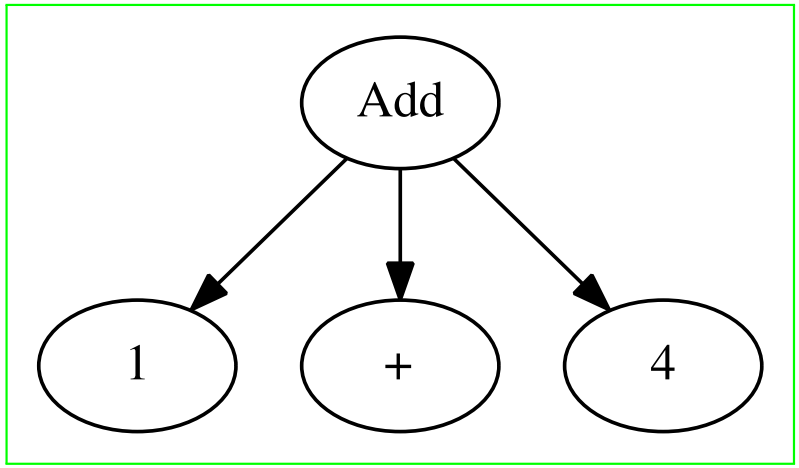
Forest minimization

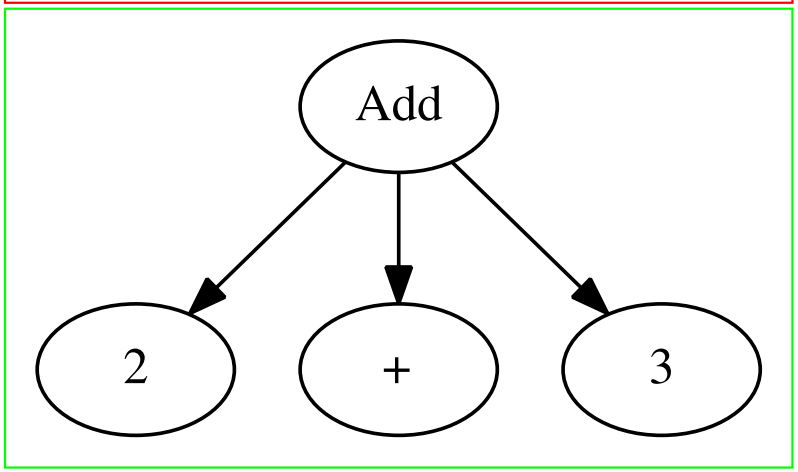


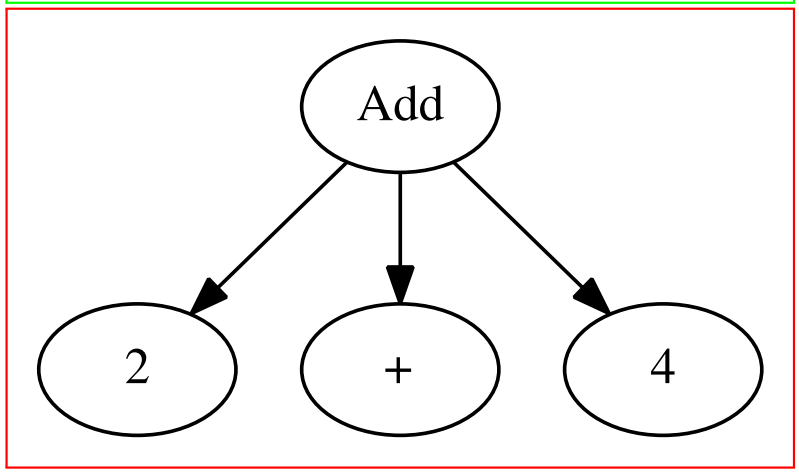
Forest minimization









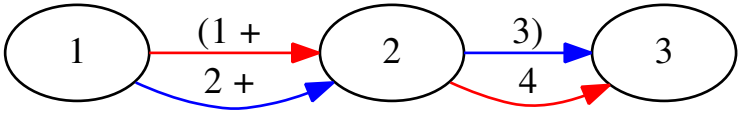


Forest filtration

- Runtime filtration in each vertice with multiple input edges
  - Results with unique parser states
  - Minimal set of paths which contains all edges
- Why not static filtration of input graph?
- We can not predict path correctness

Static selection problem

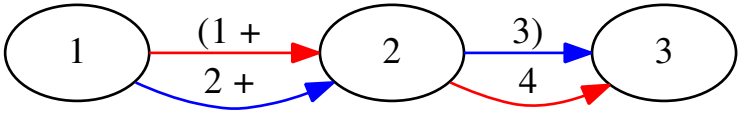
- Possible result of static paths selection:



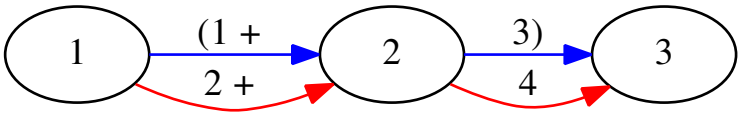
  - All selected paths are incorrect
  - No trees in result

Static selection problem

- Possible result of static paths selection:



  - All selected paths are incorrect
  - No trees in result
- Seems we should select other set of paths.



  - 2 correct trees
  - All variables are used

Conclusion

Described algorithm was implemented and used for migration of production system

- Full processing in 2 hours
- Fully processed expressions: 2181 $\rightarrow$ 2253
- Finished by timeout (not processed): 253 $\rightarrow$ 42
- It is possible to use ideas of GLR to improve our algorithm

Contact Information

- Grigorev Semyon: Semen.Grigorev@jetbrains.com
- YaccConstructor: http://recursive-ascent.googlecode.com