



Диагностика синтаксических ошибок в динамически формируемом коде

В рамках проекта лаборатории JetBrains

Автор: Рустам Шухратуллович Азимов, 444 группа

Научный руководитель: м.и.т., ст.пр. С.В. Григорьев

Рецензент: программист "ИнтеллиДжей Лабс" Д.А. Авдюхин

Санкт-Петербургский государственный университет
Кафедра системного программирования

2 июня 2016г.

- Динамический SQL

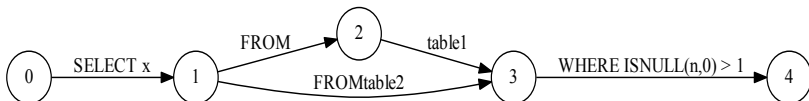
```
IF @X = @Y
    SET @TBL = ' table1 '
ELSE
    SET @TBL = 'table2 '
SET @S = 'SELECT x FROM' + @TBL + 'WHERE ISNULL(n,0) > 1'
EXECUTE (@S)
```

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

- Alvor, Java String Analyzer, PHP SA, Kyung-Goo Doh et al.
 - ▶ Имеется диагностика синтаксических ошибок, но плохо расширяемы и не строят структурного представления кода
- Ослабленный синтаксический анализ регулярной аппроксимации динамически формируемого кода в рамках проекта YaccConstructor
 - ▶ Строит деревья вывода, хорошо расширяем, но синтаксические ошибки игнорируются
 - ▶ **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

Пример

- IF @X = @Y
 SET @TBL = ' table1 '
ELSE
 SET @TBL = 'table2 '
SET @S = 'SELECT x FROM' + @TBL + 'WHERE ISNULL(n,0) > 1'
EXECUTE (@S)
- Множество значений:
{ 'SELECT x FROM table1 WHERE ISNULL(n,0) > 1' ;
 'SELECT x FROMtable2 WHERE ISNULL(n,0) > 1' }
- Аппроксимация:



Постановка задачи

Цель: разработка механизма диагностики ошибок в ослабленном синтаксическом анализе регулярной аппроксимации динамически формируемого выражения

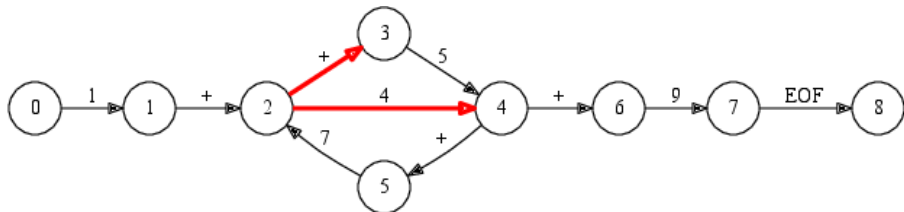
Задачи:

- Определить понятие синтаксической ошибки в терминах регулярной аппроксимации динамически формируемого выражения
- Разработать механизм диагностики ошибок в синтаксическом анализе регулярной аппроксимации динамически формируемого выражения
- Доказать корректность
- Реализовать предложенный механизм в рамках проекта YaccConstructor
- Провести экспериментальное исследование

Понятие синтаксической ошибки

- На вход вместо строки — граф ДКА
- Вместо лексем — ребра, нагруженные лексемами
- Часть прочитанного входа не префикс строки, а путь в графе из начальной вершины (далее префикс входного графа)
- Префикс входного графа корректен для рассматриваемой грамматики G , если строка, образованная лексемами данного префикса, является корректным префиксом в грамматике G
- Синтаксической ошибкой является ребро, делающее хотя бы один корректный префикс входного графа не корректным

Грамматика: $s ::= \text{Num} + s \mid \text{Num}$

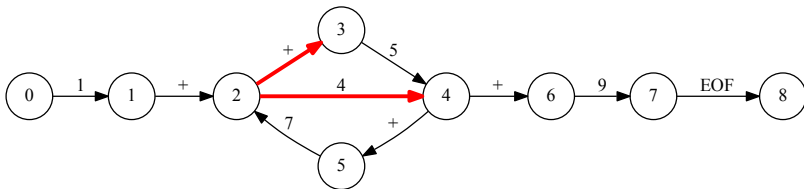


Механизм диагностики ошибок

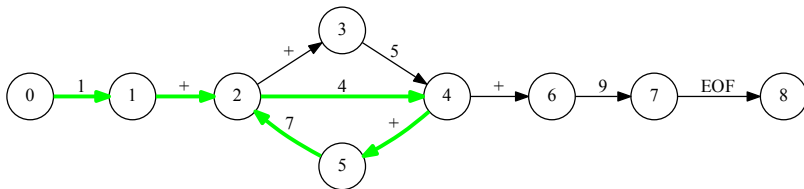
- **Построение префиксов:** во время синтаксического анализа строит для каждого LR-состояния все возможные корректные префиксы входного графа, ведущие в данное состояние
- **Диагностика ошибок:** обход в глубину по входному графу, сравнение множеств построенных префиксов
- **Результат:**
 - ▶ *errors* — гарантированные ошибки, *errors*[*e*] — гарантированно корректные префиксы, для ребра *e*
 - ▶ *probErrors* — возможные ошибки, *probErrors*[*e*] — возможно корректные префиксы для ребра *e*
 - ▶ *errors* \cup *probErrors* — аппроксимация сверху всех ошибок, *errors*[*e*] \cup *probErrors*[*e*] — аппроксимация сверху всех корректных префиксов для ребра *e*

Пример работы алгоритма

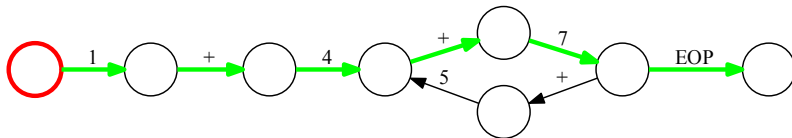
- Грамматика:
 $s ::= \text{Num} + s \mid \text{Num}$
- Входной граф ДКА:



Пример работы алгоритма: префиксы



Префиксы:



Корректность

Теорема (Корректность алгоритма построения префиксов)

Для любого LR-состояния строятся все корректные префиксы входного графа, ведущие в данное состояние, и только они.

Теорема (Корректность алгоритма диагностики ошибок)

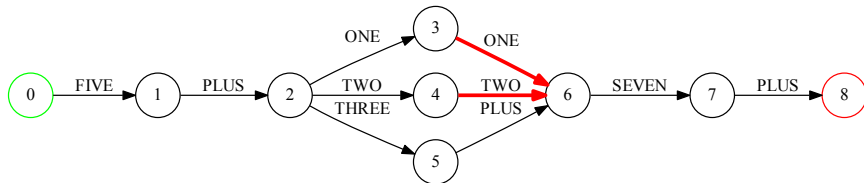
Все ребра из построенного множества $errors$ — ошибочные, а множество $errors \cup probErrors$ — аппроксимация сверху всех ошибочных ребер.

Теорема (Корректность алгоритма диагностики ошибок)

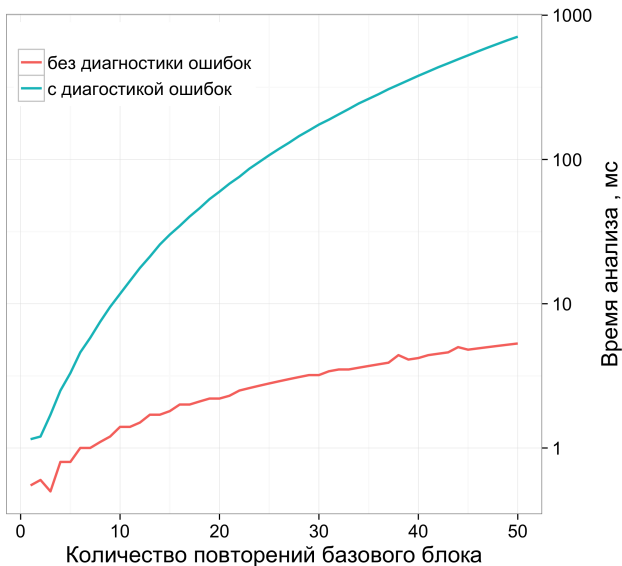
Для каждого ребра e , все префиксы входного графа из $errors[e]$ — корректные, но становятся некорректными при добавлении в конец ребра e , а множество $errors[e] \cup probErrors[e]$ — аппроксимация сверху всех таких префиксов.

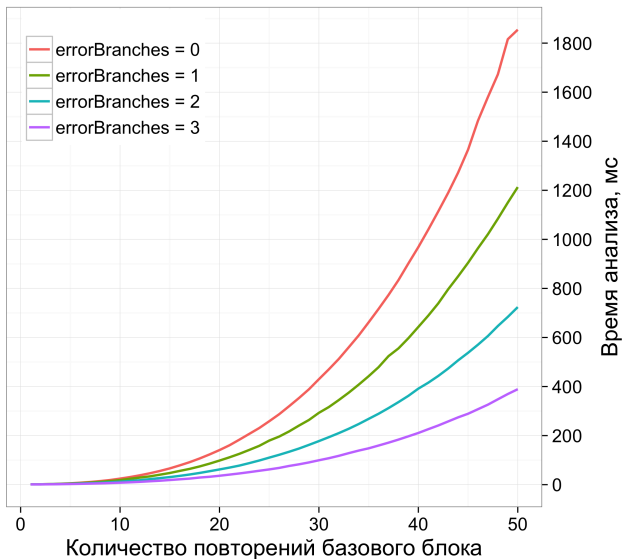
Эксперимент

- Грамматика:
 $s ::= s \text{ PLUS } n$
 $n ::= \text{ONE} \mid \text{TWO} \mid \text{THREE} \mid \text{FOUR} \mid \text{FIVE} \mid \text{SIX} \mid \text{SEVEN}$
- Базовый блок с количеством ветвлений $height = 3$ и с количеством ошибочных веток $errorBranches = 2$



Эксперимент





- Определено понятие синтаксической ошибки в терминах регулярной аппроксимации динамически формируемого выражения
- Разработан механизм диагностики ошибок в синтаксическом анализе регулярной аппроксимации динамически формируемого выражения
- Выполнена реализация предложенного механизма на языке программирования F# в рамках исследовательского проекта YaccConstructor
- Доказана корректность
- Проведено экспериментальное исследование