

Поддержка строковых операций в лексическом анализе динамически формируемого кода

Анна Явейн

Куратор: Екатерина Вербицкая

2015

■ 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)
```

■ Встроенный SQL

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

Лексический анализ и строковые операции

На входе:

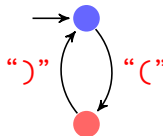
```
string res = "";  
for (int i = 0; i < l; ++i)  
    res = "(" + res;  
do_something_with(res);
```

Множество значений:

{ "", "(", ..., "("^N }

Аппроксимация:

$("(")^*$



- Конкатенация.

Что было

- Конкатенация.
- Объединение.

- Конкатенация.
- Объединение.
- Пересечение.

- Конкатенация.
- Объединение.
- Пересечение.
- Избыточная аппроксимация для `replace`.

Зачем нужен replace

На входе:

```
var col = "a_b_c";  
replace(col, "_", " ");  
db.execute("select" + col +  
           "from tbl");
```

"select a_b_c from tbl" - identifier

"select a b_c from tbl" - identifier + alias

"select a b c from tbl" - error

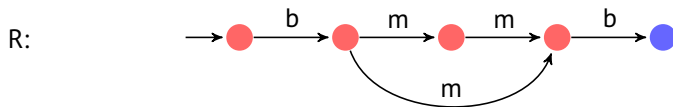
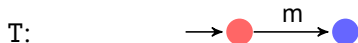
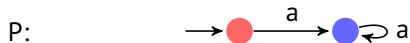
Избыточная аппроксимация для replace

$$\text{replace}(F, P, T) = R$$

$\omega \in R$ тогда и только тогда, когда

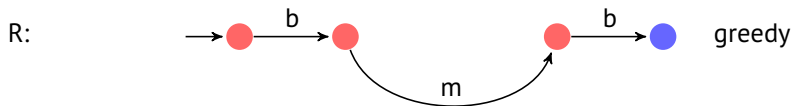
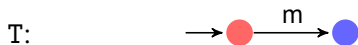
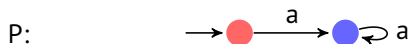
- $\exists \omega' \in F: \omega' = \omega_1 x_1 \omega_2 x_2 \dots \omega_k x_k \omega_{k+1}, x_i \in P.$
- ω_i не содержат подстроки из $P.$
- $\omega = \omega_1 c_1 \omega_2 c_2 \dots \omega_k c_k \omega_{k+1}, c_i \in T.$

Избыточная аппроксимация для replace



- Жадная (**greedy**) семантика.

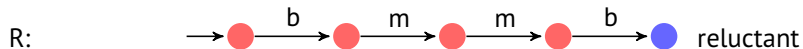
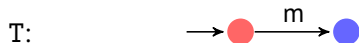
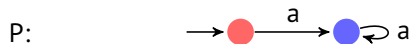
Greedy replace



Возможные сужения аппроксимации

- Жадная (**greedy**) семантика.
- Ленивая (**reluctant**) семантика.

Reluctant replace

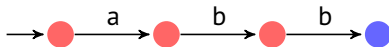


Возможные сужения аппроксимации

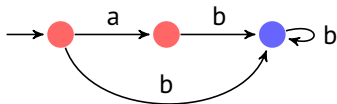
- Жадная (**greedy**) семантика.
- Ленивая (**reluctant**) семантика.
- Самое левое совпадение (**leftmost**).

Leftmost replace

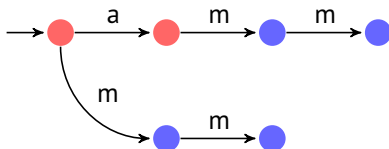
F:



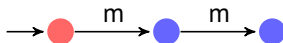
P:



R:

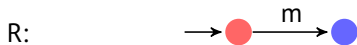
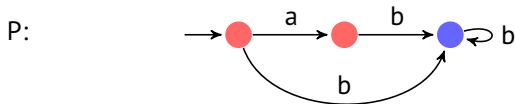
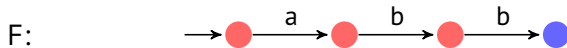


R:



leftmost

Leftmost replace



leftmost + greedy



leftmost + reluctant

Возможные сужения аппроксимации

- Жадная (**greedy**) семантика.
- Ленивая (**reluctant**) семантика.
- Самое левое совпадение (**leftmost**).
- Замена только первого вхождения (**replaceFirst**).

Что дальше?

F: aba a

P: aba | abaa

T: m

R: m a (PCRE)

R: m (C++ ?!)

- Новый язык: F#.
- Регулярные выражения.
- Автоматы и трансдюсеры.
- Академический характер задачи.
- Найденные исследования оказались неприменимы в моем случае.

Спасибо за внимание.

Репозиторий на GitHub:

`YaccConstructor/QuickGraph`

`anya.yaveyn@yandex.ru`