

YaccConstructor

Задачи на весенний семестр 2016

Автор: Семён Григорьев

Лаборатория языковых инструментов JetBrains
Санкт-Петербургский государственный университет
Математико-механический факультет

8 февраля 2016г.

- Исследования в области лексического и синтаксического анализа
- Открытый исходный код
 - ▶ <https://github.com/YaccConstructor>
- Основной язык разработки — F#

Требования к знаниям и навыкам

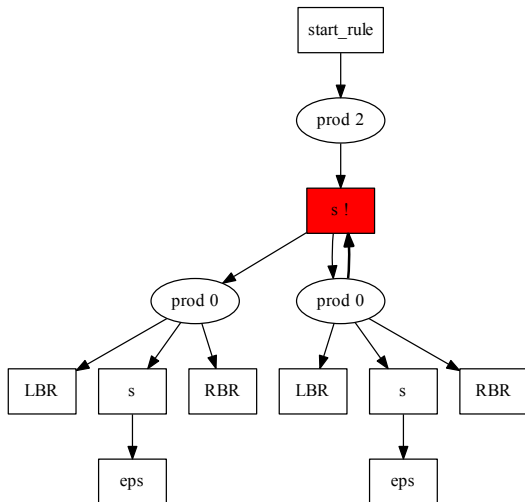
- Знакомство с функциональным программированием (F# или OCaml)
- Знания в области синтаксического анализа
- Знания в области техник трансляции
- Умение читать научные статьи, чужой код, молитвы

Анализ динамически формируемого кода

Динамически формируемый код

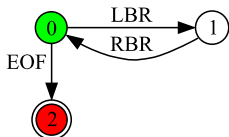
```
string res = "";  
for(i = 0; i < l; i++)  
    res = "()" + res;  
use(res);
```

Лес разбора (SPPF)



Аппроксимация

("()")*



Грамматика

```
start ::= s  
s ::= LBR s RBR s  
s ::= ε
```

Задача: применение анализа строковых выражений для JavaScript eval

- Цель: трансляция стандартного eval в "безопасный"
 - ▶ Information Flow Analysis for a Dynamically Typed Functional Language with Staged Metaprogramming
 - ▶ Martin Lester. Analysing Eval using Staged Metaprogramming
- Исследовательская задача: диплом, публикации
- Разбивается на 2 подзадачи
 - ▶ Получение аппроксимации
 - ▶ Трансляция SPPF в "безопасный eval"

Задача: использование SPPF в абстрактном синтаксическом анализе

- Абстрактный синтаксический анализ — один из подходов к анализу динамически формируемого кода
 - ▶ K. G. Doh, H. Kim, D. A. Schmidt. Static Validation of Dynamically Generated HTML Documents Based on Abstract Parsing and Semantic Processing
 - ▶ E. Verbitskaia, S. Grigorev, D. Avdyukhin. Relaxed Parsing of Regular Approximations of String-Embedded Languages
- Реализовать вычисление семантики по статьям
- Сравнить с нашим подходом
- А можно ли использовать SPPF
- Диплом, публикации

Средства для сертификационного
программирования

или

$$F\# + F^* = ?$$

Сертификационное программирование

- F^* (<https://www.fstar-lang.org/tutorial/>)
- Coq
- Agda
- ...

```
1 val sort: l:list int ->
2     Tot (m:list int{sorted m
3         /\ (forall i. mem i l = mem i m)})
4     (decreases (length l))
5 let rec sort l = match l with
6 | [] -> []
7 | pivot::tl ->
8     let hi, lo = partition (cmp pivot) tl in
9     let l' = append (sort lo) (pivot::sort hi) in
10    dedup l'
```

Задача: объединение $F\#$ и F^*

- $F\# + F^* = F\#^*$
- Парсер
- Транслятор
- Разбивается на подзадачи (до Т человек)
- Результаты можно будет опубликовать

Задача: поддержка $F\#^*$ в Microsoft Visual Studio

- Поддержать в: модели проекта, редакторе, отладчике
- Ошибки
- Разбивается на подзадачи (до Т человек)
- Результаты можно будет опубликовать

Задача: межъязыковое взаимодействие

- Использовать функции, написанные на F^* в $F\#(.NET)$
- Использовать функции, написанные на $F\#(.NET)$ в F^*
- Сохранить типизацию/вывод типов
- На диплом: реализовать и сравнить оба подхода
- Разбивается на подзадачи (до 2 человек)
- Результаты можно будет опубликовать

- Почта: `rsdpisuy@gmail.com`
- Исходный код YaccConstructor:
`https://github.com/YaccConstructor`
- Google+ сообщество: `https://goo.gl/DuPWkM`