Поддержка произвольных конечных автоматов в алгоритме синтаксического анализа регулярной аппроксимации кода на встроенных языках

YaccConstructor

Выступающий: И. Шугаепов
Руководитель: Е. Вербицкая

Лаборатория языковых инструментов JetBrains

21 декабря 2015

Сфера применимости алгоритма

Встроенный код

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

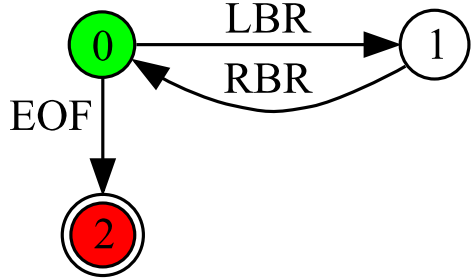
Возможные значения

{"", "()", "()()", ..., "()"^l}

Аппроксимация

("()")*

Соответствующий КА



Алгоритм

- **Вход:** КС-грамматика G и конечный автомат над алфавитом терминалов из G
- **Выход:** конечное представление множества деревьев, соответствующих всем корректным цепочкам, принимаемым входным автоматом

Цели и задачи

Целью работы было снятие ограничений на входную структуру данных (входной автомат). То есть:
Детерминированный автомат без ϵ-переходов $\Rightarrow$ произвольный конечный автомат.
Задачи:

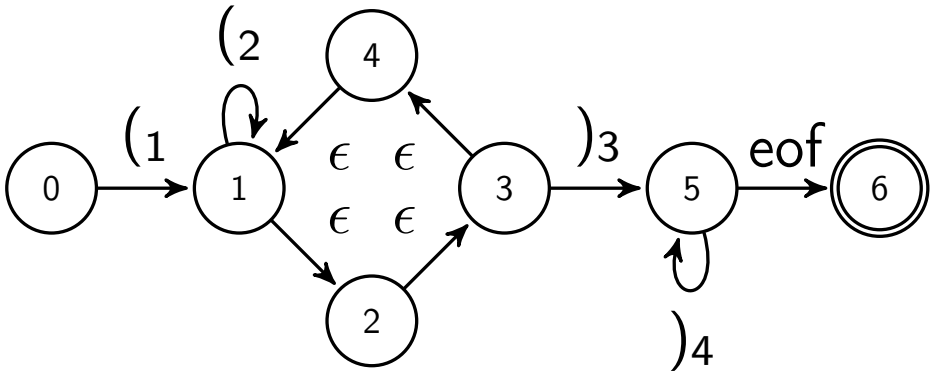
- Поддержать множественные исходящие ребра с одинаковыми метками
- Поддержать множественные начальные вершины
- Поддержать множественные конечные вершины
- Поддержка ϵ переходов

Теория

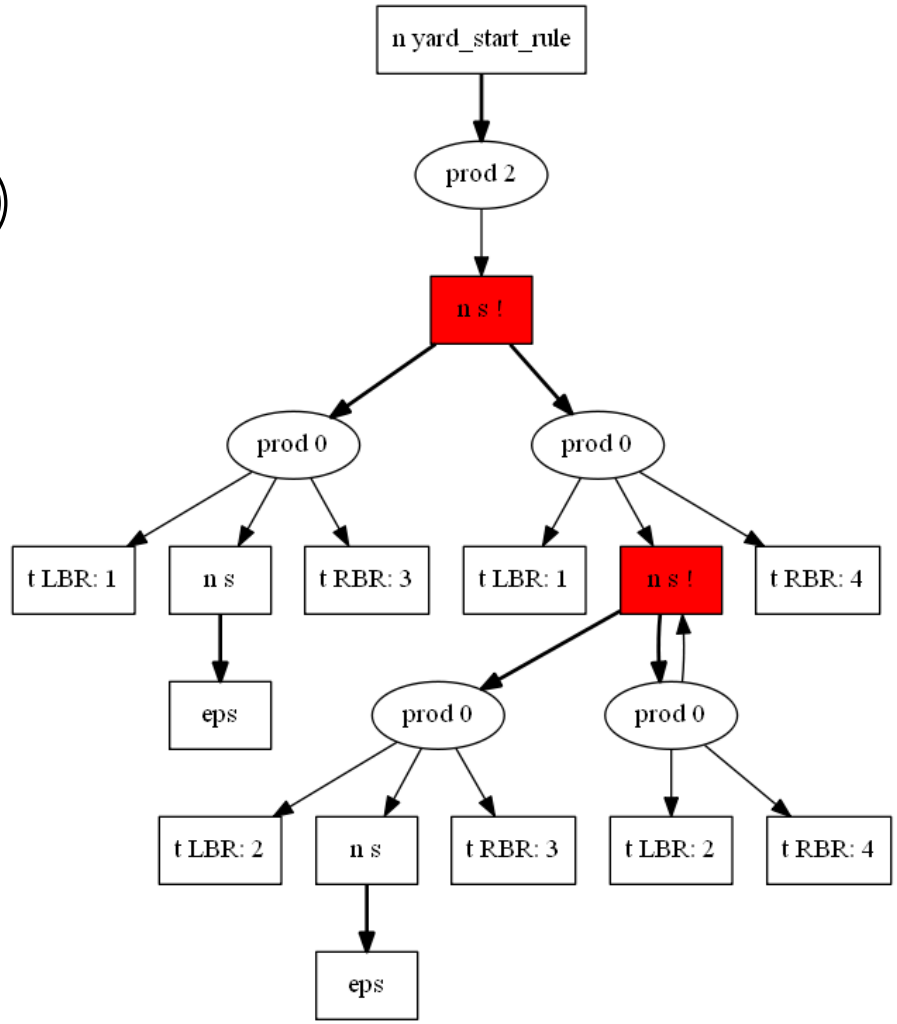
- Left-to-right Rightmost derivation parser (LR)
- Generalized LR (Tomita)
- Right Nulled Generalized LR

Пример 1

Входной автомат:

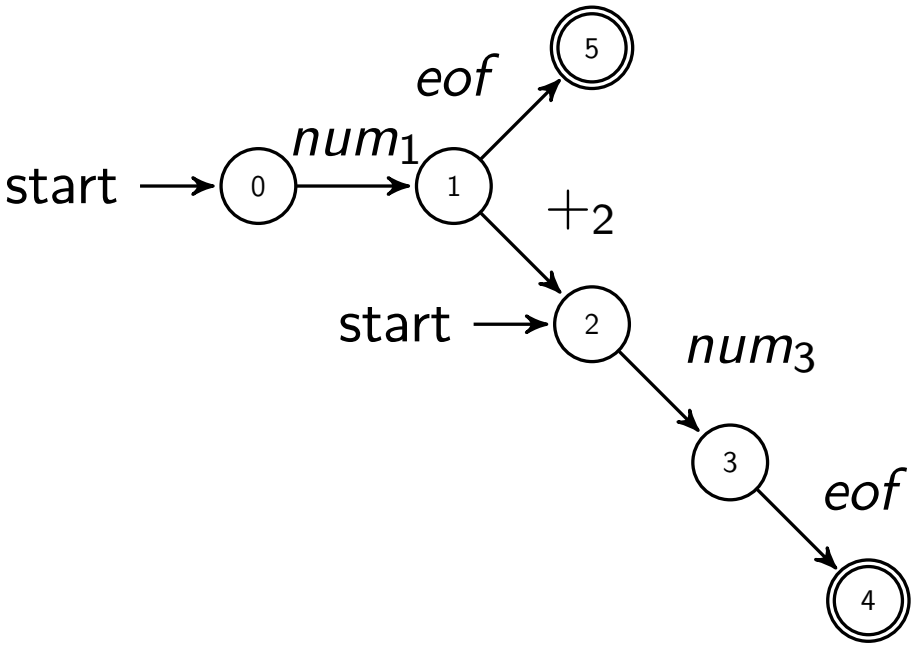


SPPF:

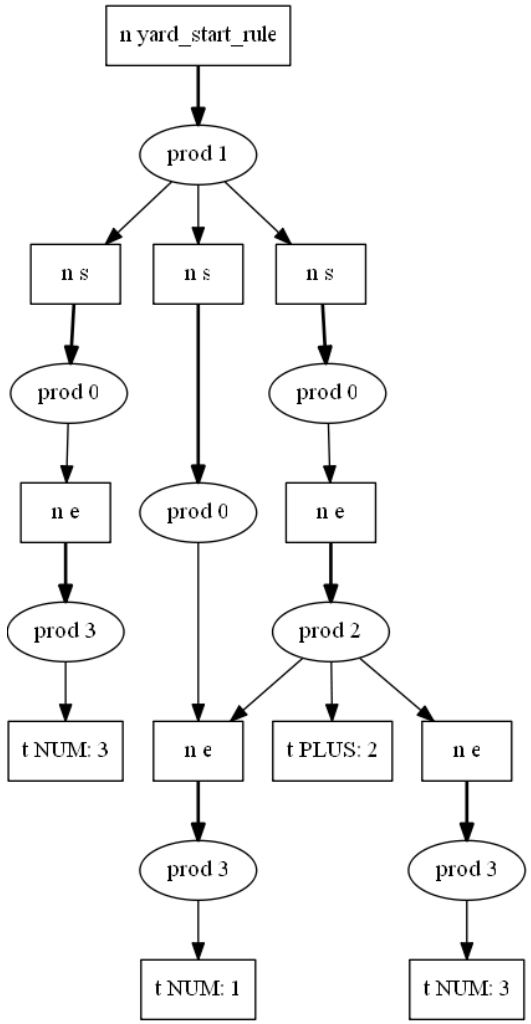


Пример 2

Входной автомат:



SPPF:



Результаты

- Реализована поддержка:
  - Множественных исходящих ребра с одинаковыми метками
  - Множественных начальных вершины
  - Множественных конечных вершины
- Частично реализовано:
  - Поддержка ϵ переходов

Исходный код YaccConstructor:
`https://github.com/YaccConstructor/YaccConstructor`
Ветка iss103 eps