



# Генератор анализаторов с поддержкой неоднозначных атрибутивных EBNF-грамматик в среде .NET

**Автор:** Григорьев Семён Вячеславович  
**Научный руководитель:** к.ф.-м.н. А.С. Лукичёв

Санкт-Петербургский государственный университет  
Математико-Механический факультет  
Кафедра системного программирования

23 марта 2011г.

Реинжиниринг программного обеспечения:

- Упрощение создания и сопровождения грамматик
  - ▶ Нет необходимости задавать однозначную контекстно-свободную
  - ▶ Нет десятков конфликтов при одном изменении
- Работа с диалектами одного языка
  - ▶ Задание общей грамматики
  - ▶ Автоматическое определение диалекта

# Цели и задачи

## Цель

Разработка генератора синтаксических анализаторов для среды .NET со следующими свойствами:

- Работа с произвольными контекстно-свободными грамматиками
- Поддержка EBNF-грамматик
- Поддержка семантических вычислений

## Задача

Реализовать поддержку s-атрибутных и l-атрибутных грамматик.

GLR-анализатор предназначен для работы с произвольной  
(в том числе неоднозначной!) КС грамматикой

- Для однозначных грамматик работает за линейное время
- $O(n^3)$  в худшем случае

Рассмотренные подходы:

- Алгоритм Эрли
- Алгоритм Томиты
- Рекурсивно-восходящий алгоритм

Конструкции регулярных выражений в правых частях правил.

Пример грамматики:

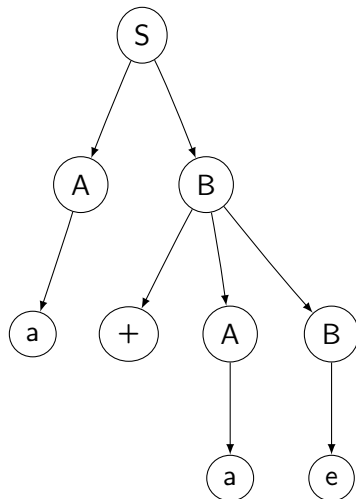
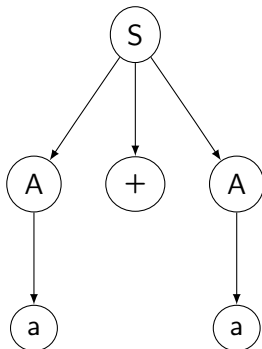
- $S \rightarrow A(+A)^*$
- $A \rightarrow a$

Преобразованная грамматика:

- $S \rightarrow AB$
- $A \rightarrow a$
- $B \rightarrow +AB$
- $B \rightarrow \varepsilon$

Входная цепочка:  $a+a$

Ожидания пользователя:      Результат:

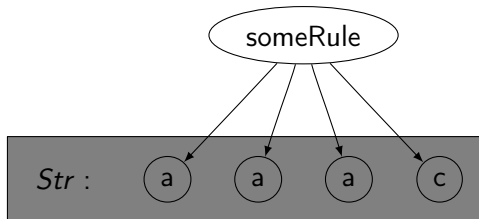


# Вычисление атрибутов

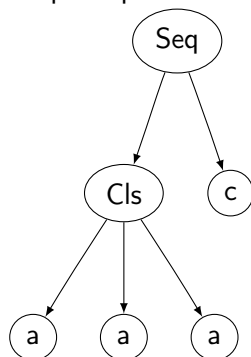
Правило грамматики:

```
someRule : val1 = (a {action1})* val2 = c {someFunc val1 val2};
```

Узел дерева вывода:

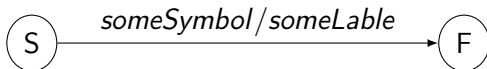


Дерево разбора *Str*:



# Вычисление атрибутов

- Конечный автомат с помеченными переходами:



- Трасса автомата:

```
[(SeqS, 1);  
  (ClsS, 1);  
    (SeqS, 2); (LeafS, 4); 'a'; (LeafE, 4); ... (SeqE, 2);  
  (ClsE, 1);  
  (LeafS, 4); 'c'; (LeafE, 4);  
  (SeqE, 1)]
```

- По трассе строится дерево разбора.



Для генератора GLR-анализаторов с поддержкой EBNF-грамматик без преобразования реализован механизм поддержки l-атрибутных и s-атрибутных грамматик:

- генератор action-кода;
- механизм вычисления атрибутов;

# Используемые продукты Microsoft

При разработке используются следующие продукты Microsoft:

- Microsoft Visual Studio 2010 – среда разработки
- F# – язык разработки и целевой язык генератора
- F# PowerPack – FsLex и FsYacc.

# Заключение

Данная разработка ведётся в рамках проекта кафедры системного программирования Математико-Механического факультета СПбГУ YaccConstructor. Исходный код и дополнительную информацию по проекту можно найти на сайте <http://code.google.com/p/recursive-ascent>.