



Генератор синтаксических анализаторов для
неоднозначных контекстно-свободных грамматик
Квалификационная работа бакалавра

Автор: Григорьев Семён Вячеславович
Научный руководитель: к.ф.-м.н. А.С. Лукичёв
Рецензент: д.ф.-м.н., проф. Б.К. Мартыненко

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

10 июня 2010г.

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

Задача

Разработать генератор синтаксических анализаторов со следующими свойствами:

- Работа с произвольными контекстно-свободными грамматиками
- Поддержка EBNF-грамматик
- Поддержка s-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:

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

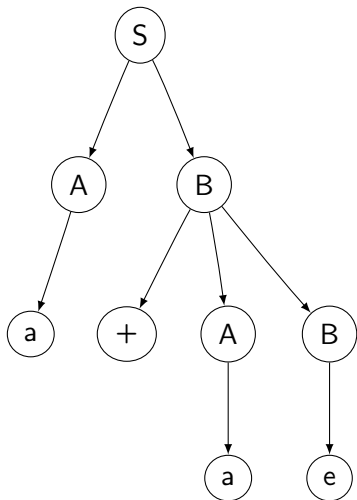
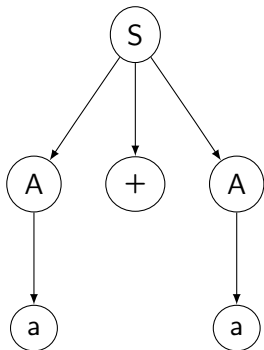
Преобразованная грамматика:

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

Входная цепочка: $a+a$

EBNF-грамматики

Ожидания пользователя: Результат:

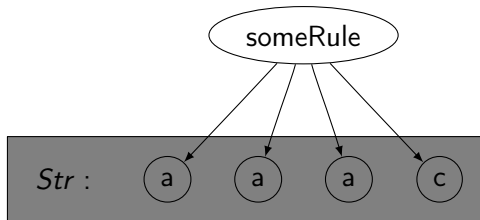


Вычисление атрибутов

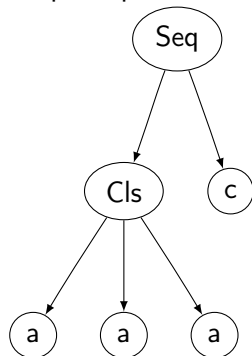
Правило грамматики:

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

Узел дерева вывода:

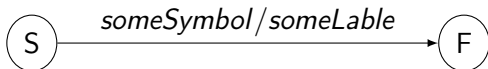


Дерево разбора *Str*:



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.

- Язык реализации – F#.
- Фронтенд – YARD – разработка кафедры системного программирования.

Реализован прототип GLR-анализатора, обладающий следующими свойствами:

- по однозначной LR-грамматике строится анализатор с линейной сложностью;
- по неоднозначной грамматике строится анализатор, возвращающий все возможные деревья вывода для данной входной цепочки;
- реализует поддержку EBNF-грамматик без их преобразования;
- реализует вычисление s -атрибутов;

- Участие в конкурсе-конференции "Технологии Microsoft в теории и практике программирования"
 - ▶ тезисы опубликованы в сборнике материалов конкурса-конференции;
 - ▶ диплом второй степени;
 - ▶ рекомендация к публикации в журнале, входящем в список ВАК;