

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №1. Структурирование формальных языков с помощью нормальной формы Бекуса-Наура

Разработчик учебно-методического материала: кандидат технических наук, доцент Рыбанов А.А.

Цели практического занятия:

- изучить модель описания грамматик языков в нотации Бекуса-Наура (БНФ);
- ознакомиться с методикой построения формы Бекуса-Наура - одной из форм записи грамматики языка программирования;
- получить навыки практического применения генератора RSyntaxTree для построения синтаксического дерева вывода.

1 Определения

В любом языке – естественном или искусственном – можно выделить три составляющие: лексику, синтаксис и семантику.

Определение 1. Лексика – это правила построения слов языка.

Определение 2. Синтаксис (грамматика языка) — это совокупность правил, согласно которым строятся допустимые в данном языке конструкции.

Определение 3. Семантика — смысловая сторона языка – она соотносит единицы и конструкции языка с некоторым внешним миром, для описания которого язык используется.

Грамматика языка программирования является формальным описанием его синтаксиса или формы, в которой записаны отдельные предложения программы или вся программа. Грамматика не описывает семантику или значения различных предложений. Информация о семантике содержится в программах генерации объектного кода. В качестве иллюстрации разницы между синтаксисом и семантикой рассмотрим два предложения:

$$I := J + K$$

и

$$I := X + V$$

, где X и V являются действительными переменными, а I, J, K -целыми переменными. Эти два предложения имеют одинаковый синтаксис. Оба являются операторами присваивания, в которых присваиваемое значение определяется выражением, состоящим из двух имен переменных, разделенных оператором сложения. Однако семантика этих двух предложений совершенно различна:

- первое предложение говорит о том, что переменные в выражении должны быть сложены с использованием целых арифметических операций, а результат сложения должен быть присвоен переменной *I*.

- второе предложение задает сложение с плавающей точкой, результат которого должен быть преобразован в целое число перед присваиванием.

Очевидно, что эти два предложения будут скомпилированы в совершенно различные последовательности машинных команд, хотя их грамматическое описание одинаково. Различия между ними проявятся на этапе генерации объектного кода.

Для описания формального языка необходим другой язык, с помощью которого будут создаваться языковые конструкции.

Определение 4. Язык, средствами которого производится описание грамматики называется метаязыком.

Метаязык должен обеспечивать как описание структурных единиц языка и правил объединения их в допустимые предложения, так и содержательную (смысловую) сторону языковых конструкций.

Существуют различные способы записи синтаксических правил, что в основном определяется условными обозначениями и ограничениями на структуру правил, принятыми в используемых метаязыках.

2 Нормальная форма Бекуса-Наура (БНФ)

Нормальная форма Бэкуса-Наура (Backus J.W., Naur P.) – один из наиболее распространенных способов описания синтаксиса языка. Этот способ был разработан для описания Алгола-60, однако, в дальнейшем он использован для многих других языков.

Рассмотрим конструкцию *<идентификатор>*, которая используется во многих языках программирования. На естественном языке определение конструкции звучит следующим образом: «*Идентификатор - это любая последовательность букв и цифр, начинающаяся с буквы*». Таким образом, возможны следующие варианты записи идентификатора: *id*, *perem1*, *ident15*, *a1b2* и т.д. Описание конструкции *<идентификатор>* в БНФ приведено в следующем примере.

Пример №1. Грамматика идентификатора.

```
<идентификатор> ::= <буква> | <идентификатор><буква>
                        | <идентификатор><цифра>
<буква> ::= a | b | c | ... | z
<цифра> ::= 0 | 1 | 2 | ... | 9
```

Видно, что при определении нетерминала $\langle \text{идентификатор} \rangle$ присутствует рекурсивность, поскольку нетерминал $\langle \text{идентификатор} \rangle$ определяется через самого себя. Элементарным идентификатором оказывается идентификатор из одной буквы.

Пример построения идентификатора:

Рассмотрим, как можно построить идентификатор "a1b2".

1. Начнем с первого правила для идентификатора. Мы можем использовать конструкцию $\langle \text{идентификатор} \rangle \langle \text{цифра} \rangle$, чтобы добавить цифру к идентификатору.

- Выбираем: $\langle \text{идентификатор} \rangle ::= \langle \text{идентификатор} \rangle \langle \text{цифра} \rangle$
- Выбираем: $\langle \text{цифра} \rangle ::= 2$
- Теперь нам нужно определить $\langle \text{идентификатор} \rangle$, который будет предшествовать цифре "2".

2. Для $\langle \text{идентификатор} \rangle$ перед цифрой "2" мы можем использовать $\langle \text{идентификатор} \rangle \langle \text{буква} \rangle$.

- Выбираем: $\langle \text{идентификатор} \rangle ::= \langle \text{идентификатор} \rangle \langle \text{буква} \rangle$
- Выбираем: $\langle \text{буква} \rangle ::= b$
- Теперь нам нужно определить $\langle \text{идентификатор} \rangle$, который будет предшествовать букве "b".

3. Для $\langle \text{идентификатор} \rangle$ перед цифрой "b" мы можем использовать $\langle \text{идентификатор} \rangle \langle \text{цифра} \rangle$.

- Выбираем: $\langle \text{идентификатор} \rangle ::= \langle \text{идентификатор} \rangle \langle \text{цифра} \rangle$
- Выбираем: $\langle \text{цифра} \rangle ::= 1$
- Теперь нам нужно определить $\langle \text{идентификатор} \rangle$, который будет предшествовать цифре "1".

4. Для $\langle \text{идентификатор} \rangle$ перед цифрой "1" мы можем использовать $\langle \text{буква} \rangle$.

- Выбираем: $\langle \text{идентификатор} \rangle ::= \langle \text{буква} \rangle$
- Выбираем: $\langle \text{буква} \rangle ::= a$

Теперь у нас есть следующая последовательность вывода цепочки "a1b2":

$$\begin{aligned} \langle \text{идентификатор} \rangle &::= \langle \text{идентификатор} \rangle \langle \text{цифра} \rangle ::= \langle \text{идентификатор} \rangle 2 ::= \\ &::= \langle \text{идентификатор} \rangle \langle \text{буква} \rangle 2 ::= \langle \text{идентификатор} \rangle b 2 ::= \\ &::= \langle \text{идентификатор} \rangle \langle \text{цифра} \rangle b 2 ::= \langle \text{идентификатор} \rangle 1 b 2 ::= \\ &::= \langle \text{буква} \rangle 1 b 2 ::= a 1 b 2 \end{aligned}$$

Синтаксическое дерево построения цепочки "a1b2" приведено на (рис. 1).

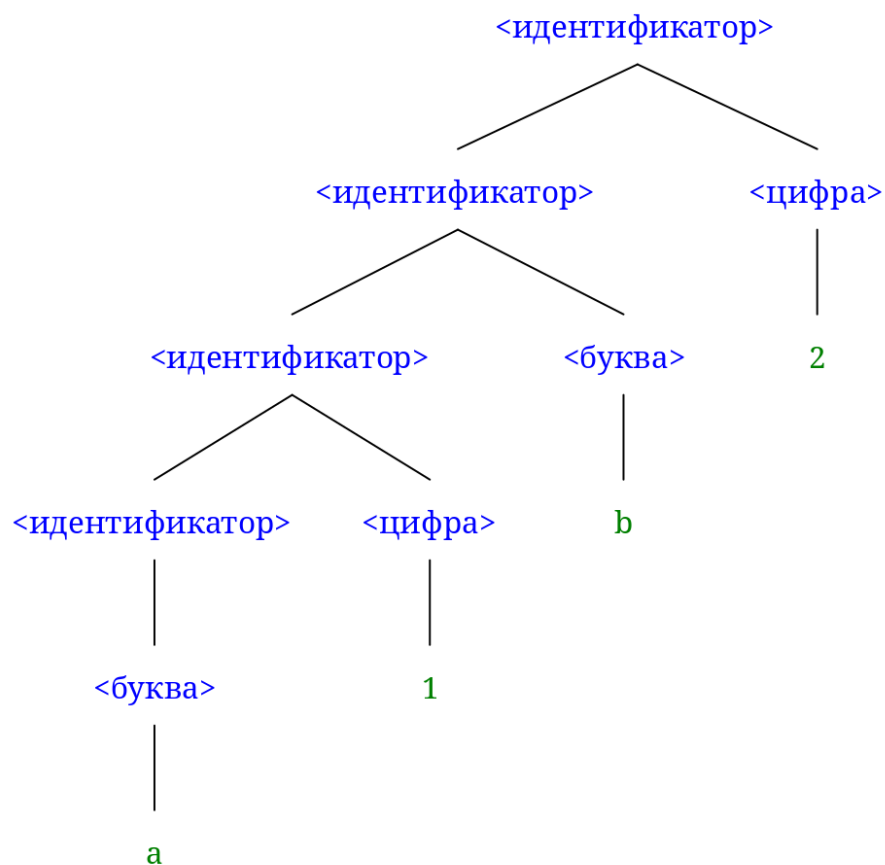


Рисунок 1 – Синтаксическое дерево вывода цепочки a1b2

3 Правила описания синтаксиса языка в нормальной форме Бекуса-Наура

Общий вид предложения в нормальной форме Бекуса-Наура выглядит следующим образом:

$$\langle \text{определяемый символ} \rangle ::= \langle \text{вариант 1} \rangle \mid \langle \text{вариант 2} \rangle$$

Для формирования предложений в нормальной форме Бекуса-Наура используются универсальные метасимволы: \langle , \rangle , $::=$, \mid .

Метасимволы \langle и \rangle называют *угловыми скобками* – они используются для обозначения нетерминального символа.

Метасимвол $::=$ читается как «*по определению есть*».

Метасимвол \mid читается как «*или*».

В предложениях, записанных в форме Бекуса-Наура, нетерминальный символ, стоящий в угловых скобках, играет роль определенной конструкции языка. В формулах Бекуса-Наура могут использоваться терминальные символы из алфавита языка, отличные от универсальных метасимволов.

Рассмотрим конструкцию в нормальной форме Бекуса-Наура, описывающей правильные скобочные последовательности.

Пример №2. Правильные скобочные последовательности.

$$\langle \text{правильная посл.} \rangle ::= () \mid (\langle \text{правильная посл.} \rangle) \mid \langle \text{правильная посл.} \rangle \langle \text{правильная посл.} \rangle$$

Описание формального языка строится из последовательности формул, каждая из которых в левой части содержит один метасимвол, обозначающий некоторую конструкцию языка. Правая часть такой формулы содержит либо перечисление метасимволов и терминальных символов языка (никаких разделителей при этом не ставится), либо совокупности перечислений, разделенных метасимволом $|$. Правая и левая часть объединяются в единую формулу метасимволом $::=$.

Приведем последовательность вывода правильной скобочной последовательности $()(())$ из предложений БНФ, представленной в примере №2:

$$\begin{aligned} \langle \text{правильная посл.} \rangle ::= & \langle \text{правильная посл.} \rangle \langle \text{правильная посл.} \rangle ::= \\ & () \langle \text{правильная посл.} \rangle ::= \\ & () (\langle \text{правильная посл.} \rangle) ::= \\ & () (()) \end{aligned}$$

Язык можно считать полностью определенным в нормальной форме Бекуса-Наура, если любой нетерминальный символ можно представить последовательностью терминальных символов. Форма Бэкуса-Наура не позволяет задавать контекстные условия. При использовании формы Бэкуса-Наура контекстные условия задаются в словесной форме.

4 Генератор построения синтаксического дерева `RSyntaxTree`

Для построения синтаксического дерева вывода терминальной цепочки по грамматике, представленной в БНФ можно использовать генератор `RSyntaxTree`. Генератор синтаксического дерева `RSyntaxTree` создан с помощью Ruby и RMagic. (рис. 2)

Пример №3. Описание синтаксического дерева вывода на предметно-ориентированном входном языке генератора `RSyntaxTree`

```
[\<формула\>
  [\<формула\>
    [\<формула\>
      [\<число\>
        [\<цифра\> 3]
```

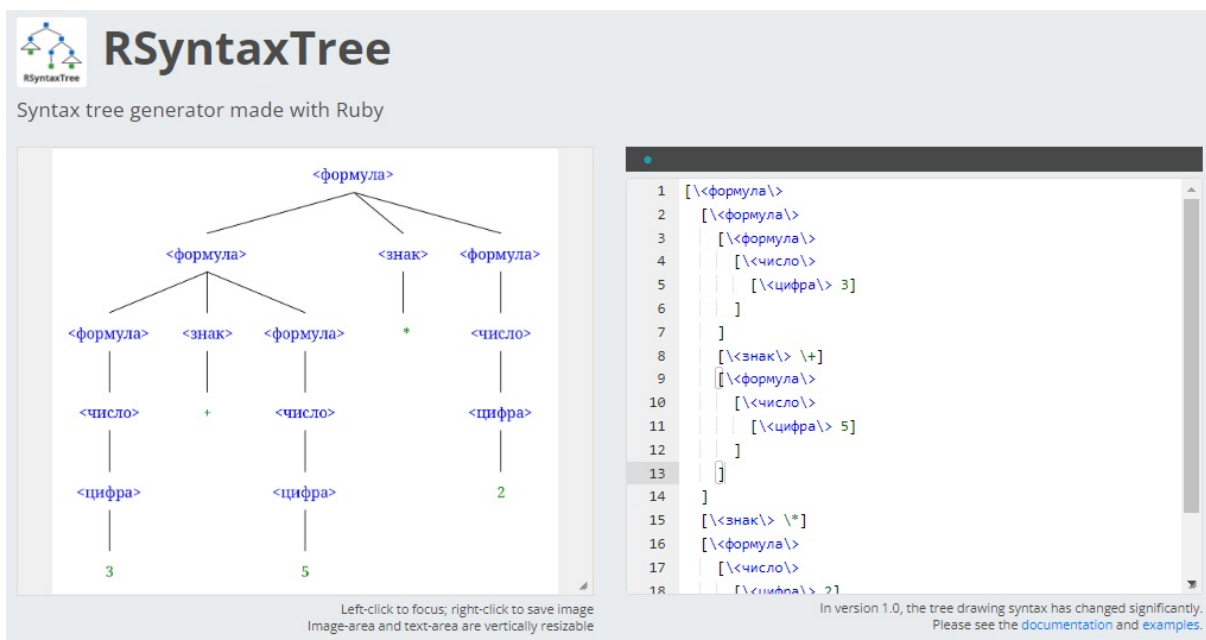


Рисунок 2 – Генератор синтаксического дерева RSyntaxTree

```

]
]
[\<знак\> \+]
[\<формула\>
  [\<число\>
    [\<цифра\> 5]
  ]
]
]
[\<знак\> \*]
[\<формула\>
  [\<число\>
    [\<цифра\> 2]
  ]
]
]
]

```

5 Примеры описаний в нормальной форме Бекуса-Наура

Рассмотрим язык простейших арифметических формул.

Пример №4. Язык простейших арифметических формул.

```

<формула> ::= (<формула>) | <число> | <формула><знак><формула>
<число>  ::= <цифра> | <цифра><число>

```

$\langle \text{цифра} \rangle ::= 0 \mid 1 \mid 2 \mid \dots \mid 9$

$\langle \text{знак} \rangle ::= + \mid - \mid * \mid /$

Синтаксическое дерево построения цепочки «3+5*2» приведено на (рис. 3).

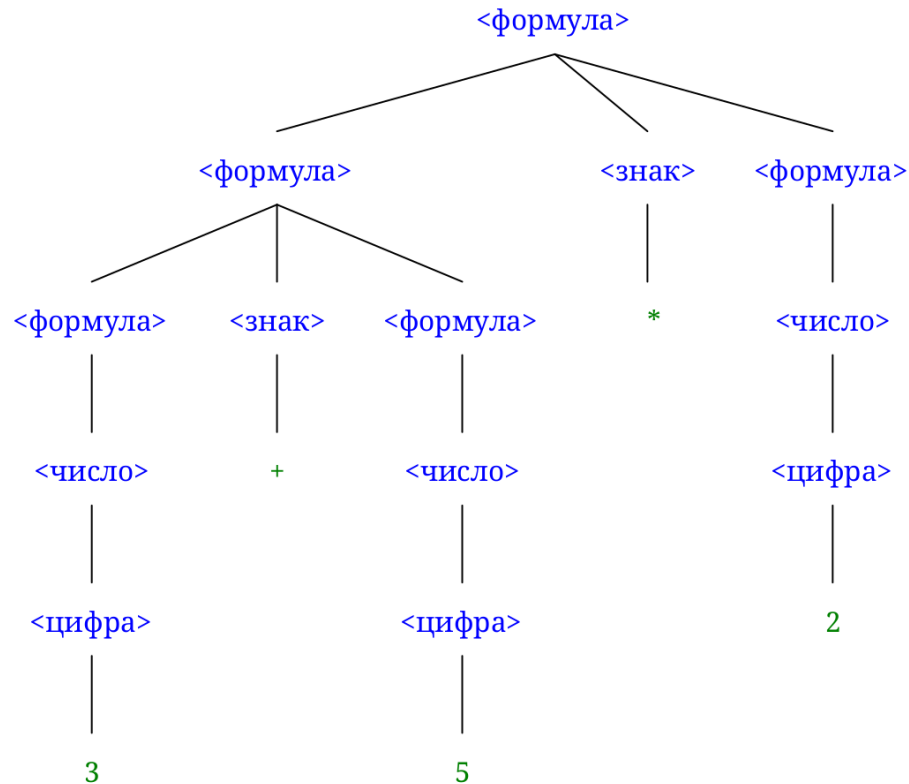


Рисунок 3 – Синтаксическое дерево вывода цепочки

Пример №5. Грамматика для списка идентификаторов.

$\langle \text{список} \rangle ::= \langle \text{идентификатор} \rangle ; \mid \langle \text{идентификатор} \rangle , \langle \text{список} \rangle$

$\langle \text{идентификатор} \rangle ::= \langle \text{буква} \rangle \mid \langle \text{идентификатор} \rangle \langle \text{буква} \rangle$

$\mid \langle \text{идентификатор} \rangle \langle \text{цифра} \rangle$

$\langle \text{буква} \rangle ::= a \mid b \mid c \mid \dots \mid z$

$\langle \text{цифра} \rangle ::= 0 \mid 1 \mid 2 \mid \dots \mid 9$

Пример №6. Грамматика для натуральных чисел

$\langle \text{натуральное число} \rangle ::= \langle \text{цифра} \rangle \mid \langle \text{цифра без ноля} \rangle \langle \text{посл. цифр} \rangle$

$\langle \text{последовательность цифр} \rangle ::= \langle \text{цифра} \rangle \mid \langle \text{цифра} \rangle \langle \text{посл. цифр} \rangle$

$\langle \text{цифра} \rangle ::= \langle \text{ноль} \rangle \mid \langle \text{цифра без ноля} \rangle$

$\langle \text{цифра без ноля} \rangle ::= 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

$\langle \text{ноль} \rangle ::= 0$

Пример №7. Грамматика для двоичных чисел

$\langle \text{двоичное число} \rangle ::= \langle \text{двоичное число} \rangle \langle \text{цифра} \rangle \mid \langle \text{цифра} \rangle$
 $\langle \text{цифра} \rangle ::= 0 \mid 1$

Пример №8. Грамматика целых чисел без знака.

$\langle \text{целое без знака} \rangle ::= \langle \text{цифра} \rangle \mid \langle \text{цифра} \rangle \langle \text{целое без знака} \rangle$
 $\langle \text{цифра} \rangle := 0 \mid 1 \mid 2 \mid \dots \mid 9$

Пример №9. Грамматика целых чисел со знаком.

$\langle \text{целое число} \rangle ::= \langle \text{целое без знака} \rangle \mid + \langle \text{целое без знака} \rangle \mid$
 $\quad \quad \quad - \langle \text{целое без знака} \rangle$
 $\langle \text{целое без знака} \rangle ::= \langle \text{цифра} \rangle \mid \langle \text{целое без знака} \rangle \langle \text{цифра} \rangle$
 $\langle \text{цифра} \rangle := 0 \mid 1 \mid 2 \mid \dots \mid 9$

6 Варианты заданий для самостоятельного выполнения

Номер варианта задания для выполнения выбирается по последней цифре зачетной книжки.

1) Для описания в БНФ из примера №1 постоить последовательность вывода и синтаксическое дерево вывода для следующего идентификатора: flid2

2) Для описания в БНФ из примера №2 постоить последовательность вывода и синтаксическое дерево вывода для следующей правильной скобочной последовательности: $(())(())$

3) Для описания в БНФ из примера №4 постоить последовательность вывода и синтаксическое дерево вывода для следующей простой арифметической формулы: $3+65/2$

4) Для описания в БНФ из примера №8 постоить последовательность вывода и синтаксическое дерево вывода для следующего целого числа без знака: 253

5) Для описания в БНФ из примера №9 постоить последовательность вывода и синтаксическое дерево вывода для следующего целого числа со знаком: +827

6) Для описания в БНФ из примера №1 постоить последовательность вывода и синтаксическое дерево вывода для следующего идентификатора: primer12

7) Для описания в БНФ из примера №2 постоить последовательность вывода и синтаксическое дерево вывода для следующей правильной скобочной последовательности: $()(())()$

8) Для описания в БНФ из примера №4 построить последовательность вывода и синтаксическое дерево вывода для следующей простой арифметической формулы: $23 * 4 / 12$

9) Для описания в БНФ из примера №8 построить последовательность вывода и синтаксическое дерево вывода для следующего целого числа без знака: 9823

10) Для описания в БНФ из примера №9 построить последовательность вывода и синтаксическое дерево вывода для следующего целого числа со знаком: -245