Небольшое лирическое отступление

Формы Бэкуса-Наура

Backus–Naur form (BNF, БНФ) – это формальная система описания синтаксиса, в которой одни синтаксические категории последовательно определяются через другие категории. БНФ используется для описания контекстно-свободных формальных грамматик, обычно используется для описания синтаксиса языков программирования, форматов документов, наборов инструкций и протоколов связи.

Термины:

Терминалы – это то, из чего состоит язык. Например, человеческий язык состоит из предложений, предложения – из слов, слова бывают подлежащими и сказуемыми. Слова же в свою очередь состоят из букв. Предложения, слова, подлежащие и сказуемые, буквы – это все терминалы. Буквы называют терминальными символами. *Нетерминал* или *нетерминальный символ* — объект, обозначающий какую-либо сущность языка (например: формула, арифметическое выражение, команда) и не имеющий конкретного символьного значения. Например, символы < и > обрамляют некоторые терминалы, знак | разделяет несколько альтернативных терминалов.

Текст может быть правильным (то-есть составленным по правилам этой грамматики) или неправильным. И основная задача теории – определить, правильный или неправильный этот текст. Правила «кодируются» с помощью своеобразных конструкций – форм Бэкуса-

Структура БНФ.

Каждая форма – это некоторая строчка.

Есть знак ::= . Он отделяет левую часть строки от правой. В правой части определения стоит БНФ-выражение, задающее структуру образца категории. Можно также встретить обозначение ≜.

Определение простейшего арифметического выражения с помощью БНФ выглядит следующим образом:

<арифметическое выражение> ::= < операнд> <знак операции> < операнд>

<операнд> :: = <идентификатор> | (<арифметическое выражение>)

<знак операции> :: = + | - | * | /

<идентификатор> :: = a | b | …| z | A | B | … | Z

Арифметическое выражение состоит из двух операндов, разделенных знаком операции. В свою очередь, любой операнд может представлять собой однобуквенный идентификатор или арифметическое выражение, заключенное в круглые скобки. Это означает, что арифметическое выражение определяется в терминах самого себя, следовательно, данное определение рекурсивно.

Более сложные варианты.

Некоторые конструкции (в правой части строки), взятые в квадратные скобки

([…]),

означают, что конструкция может отсутствовать (то-есть или присутствует один раз, или отсутствует). А если без таких скобок – значит присутствует строго один раз.
Фигурные скобки

({ … })

означают, что «обрамленная» ими конструкция повторяется некоторое (возможно нулевое) количество раз.

{/ … /}

означает, что повторение 1 или большее количество раз (то-есть ненулевое количество).

В некоторых случаях символы для нетерминалов < и >

Рассмотрим примеры:

1) Дадим определение идентификатору с помощью БНФ:

<Буква> ::= A|B|C|…|Z
<Цифра> ::= 0|1| … |9
<Идентификатор> ::= <Буква> {<Буква>|<Цифра>}

Помните? Идентификатор начинается обязательно с буквы, а дальше идет некоторое количество букв или цифр (в принятых обозначениях A, B, A1, B2C3)

Всегда есть «самый главный» терминал (и, соответственно, главная строка, в левой части которой он находится), соответствующий тому объекту, который мы исследуем. В данном случае самый главный терминал – это «Идентификатор».

2) Грамматика целых чисел без знака:

<число> ::= <цифра>|<цифра><число>

<цифра> ::= 0|1|2|…|9

Обратим внимание, что терминал «число» встречается в первой строке 2 раза. Это важный момент. Он приводит к появлению рекурсии.

3) Формула с плюсами и минусами без скобок.

<формула> ::= <число>|<формула><знак><число>

<знак> ::= +|- **(может быть и так: <знак> ::= +|-|*|/)**

Другая форма записи

<формула> ::= <слаг> {<add_знак><слаг>}

<слаг> ::= <число>

<add_знак> ::= +|-

А если и числа, и переменные, то:

<формула> ::= <слаг> {<add_знак><слаг>}

<слаг> ::= <число>|<переменная>

<add_знак> ::= +|-

4) Формула с плюсами, минусами и скобками.

<формула> ::= <ор>|<формула><знак><число>

<ор> ::= <число>|(<формула>)

<знак> ::= +|-

Во второй строке появились скобки. «ор» – это промежуточный терминал, введенный для удобства.

5) Формула с плюсами и минусами, а также с умножением и делением без скобок. Входят в формулу не только числа, но и переменные.

<формула> ::= <слаг> {<add_знак><слаг>}

<слаг> ::= <множ>{<mul_знак><множ>}

<множ> ::= <число>|<переменная>

<add_знак> ::= +|-

<mul_знак> ::= *|/

Т.е., самые крупные блоки, которые мы выделяем из строки с формулой – это слагаемые. Этим определяется самый низший приоритет операций сложения-вычитания. Отметим, что левое слагаемое не состоит из вложенных слагаемых, в

6) Формула с плюсами и минусами, с умножением и делением без скобок, а также с унарным плюсом-минусом (причем унарный знак может присутствовать перед числом или переменной лишь в единственном числе).

<формула> ::= <слаг> {<add_знак><слаг>}
<слаг> ::= <множ_со_знаком>{<mul_знак><множ_со_знаком>}
<множ_со_знаком> ::= [<add_знак>]<множ>
<множ> ::= <число>|<переменная>
<add_знак> ::= +|-
<mul_знак> ::= *|/

7) Формула с плюсами и минусами, умножением и делением и со скобками.

<формула> ::= <слаг> {<add_знак><слаг>}
<слаг> ::= <множ>{<mul_знак><множ>}
<множ> ::= <число>|<переменная>|(<формула>)
<add_знак> ::= +|-
<mul_знак> ::= *|/

8) Добавляем функции (встроенные) одного переменного (например sin и cos).

<формула> := <слаг> {<add_знак><слаг>}
<слаг> := <множ>{<mul_знак><множ>}
<множ> :=
<число>|<переменная>|(<формула>)|<Func>(<формула>)
<add_знак> := +|-
<mul_знак> := *|/
<Func> := sin|cos

<syntax> ::= <rule> | <rule> <syntax>
<rule> ::= <opt-whitespace> "<" <rule-name> ">" <opt-whitespace> "::="
<opt-whitespace> <expression> <line-end>
<opt-whitespace> ::= " " <opt-whitespace> | ""
<expression> ::= <list> | <list> <opt-whitespace> "|" <opt-whitespace> <expression>
<line-end> ::= <opt-whitespace> <EOL> | <line-end> <line-end>
<list> ::= <term> | <term> <opt-whitespace> <list>
<term> ::= <literal> | "<" <rule-name> ">"
<literal> ::= "'" <text1> "'" | """" <text2> """"
<text1> ::= "" | <character1> <text1>
<text2> ::= "" | <character2> <text2>
<character> ::= <letter> | <digit> | <symbol>
<letter> ::= "A" | "B" | "C" | "D" | "E" | "F" | "G" | "H" | "I" | "J" | "K" | "L" | "M" | "N" | "O" | "P" | "Q" | "R" | "S" | "T" | "U" | "V" | "W" | "X" | "Y" | "Z" | "a" | "b" | "c" | "d" | "e" | "f" | "g" | "h" | "i" | "j" | "k" | "l" | "m" | "n" | "o" | "p" | "q" | "r" | "s" | "t" | "u" | "v" | "w" | "x" | "y" | "z"
<digit> ::= "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"
<symbol> ::= "|" | " " | "-" | "!" | "#" | "$" | "%" | "&" | "(" | ")" | "*" | "+" | "," | "-" | "." | "/" | ":" | ";" | "<" | "=" | ">" | "?" | "@" | "[" | "\" | "]" | "^" | "_" | "`" | "{" | "|" | "}" | "~"
<character1> ::= <character> | """"
<character2> ::= <character> | "'"
<rule-name> ::= <letter> | <rule-name> <rule-char>
<rule-char> ::= <letter> | <digit> | "-"