_BuT.LTL

#Языки #LTL #Автоматное-программирование #Верификация #В-Работе #Грамматики #BNF

Язык задания и синтеза сложных автоматных моделей промышленных систем управления.

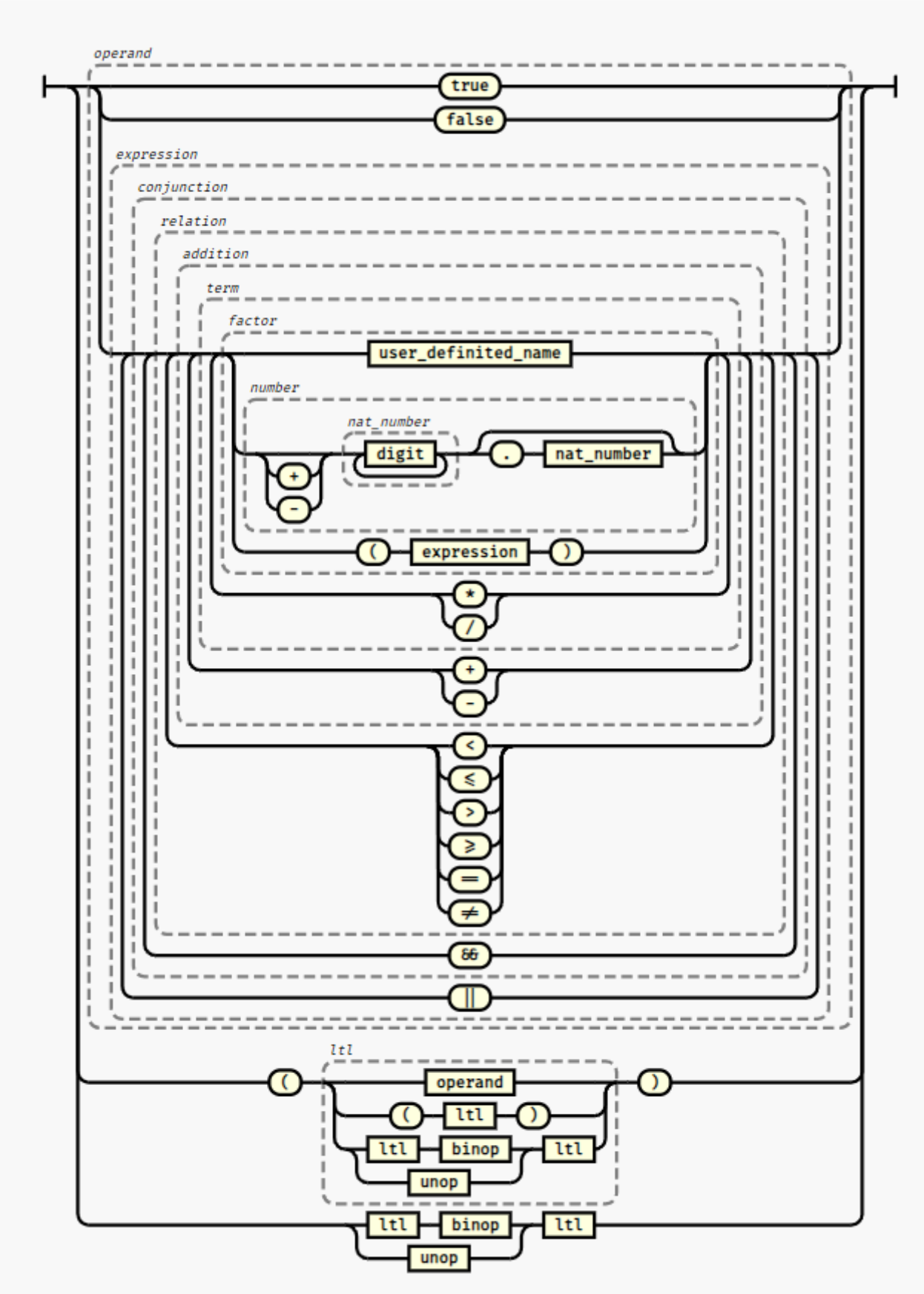
LTL формулы

В качестве базового представления формулы темпоральной логики, будет использовать синтаксис и семантика используемая в наборе утилит #SPIN. В документации к набору утилит есть описание синтаксиса. Запишем ее в расширенной нотации #BNF(#EBNF).

```
ltl = operand | "(", ltl, ")" | ltl, binop, ltl | unop, ltl;
operand = "true" | "false" | expression;
unop = "[]" | "<>" | "!";
binop = "U" | "W" | "V" | "&&" | "||" | "→" | "↔";
user_defined_name = letter, { letter | digit | "_" };
digit = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9";
nat_number = digit, { digit };
number = ["+" | "-"], nat_number, [ ".", nat_number ];
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" ;
expression = conjunction, { "||", conjunction };
conjunction = relation, { "&&", relation };
relation = addition, { ("<" | "⩽" | ">" | "⩾" | "==" | "≠"), addition
};
addition = term, { ("+" | "-"), term };
term = factor, { ("*" | "/"), factor };
factor = user_definited_name | number | "(", expression, ")";
```

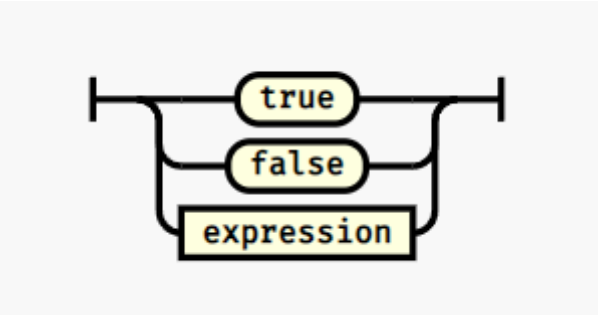
При помощи утилиты разложим грамматику на составляющие.

Полное представление грамматики



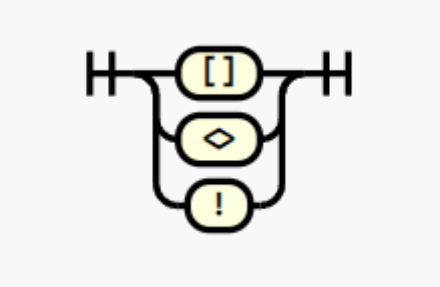
Операнд

```
operand = "true" | "false" | expression;
```



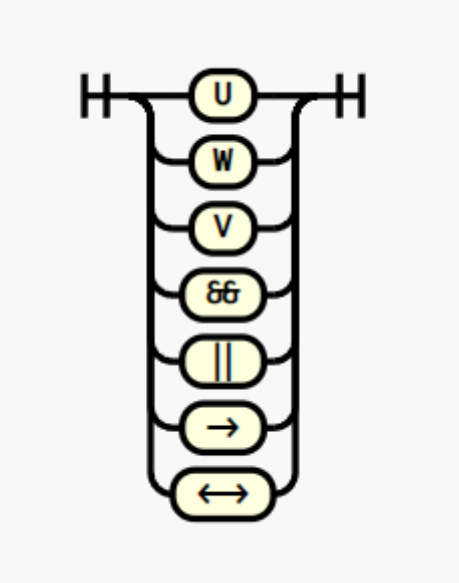
Унарная операция

```
unop = "[]" | "<>" | "!";
```



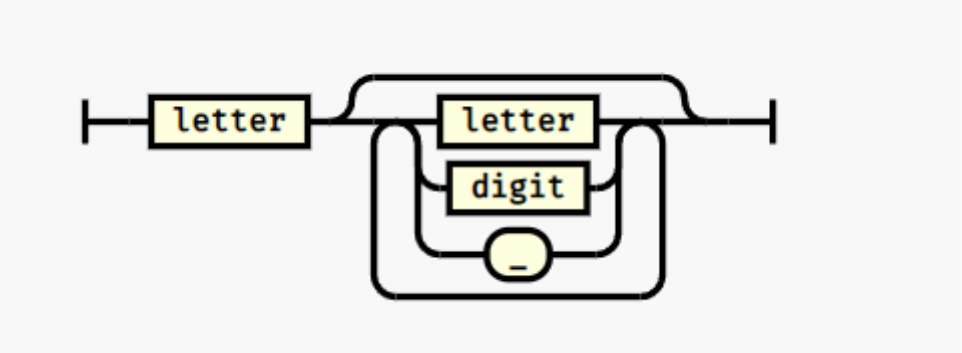
Бинарные операции

```
binop = "U" | "W" | "V" | "&&" | "||" | "→" | "⟷";
```



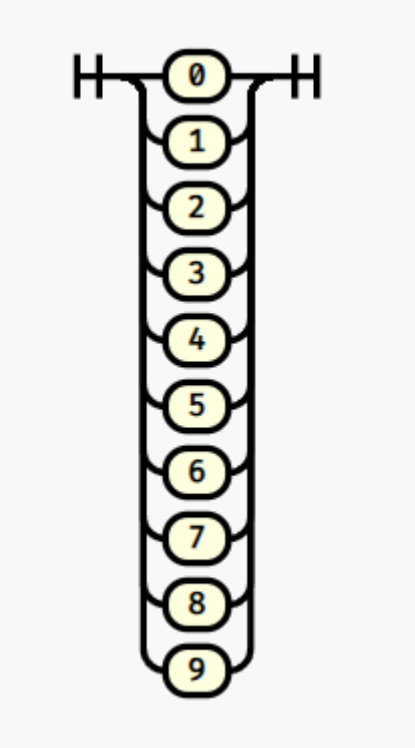
Пользовательские переменные

```
user_defined_name = letter, { letter | digit | "_" };
```



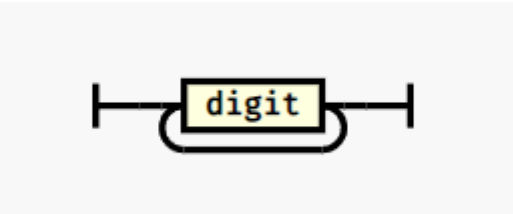
Цифры

```
digit = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9";
```



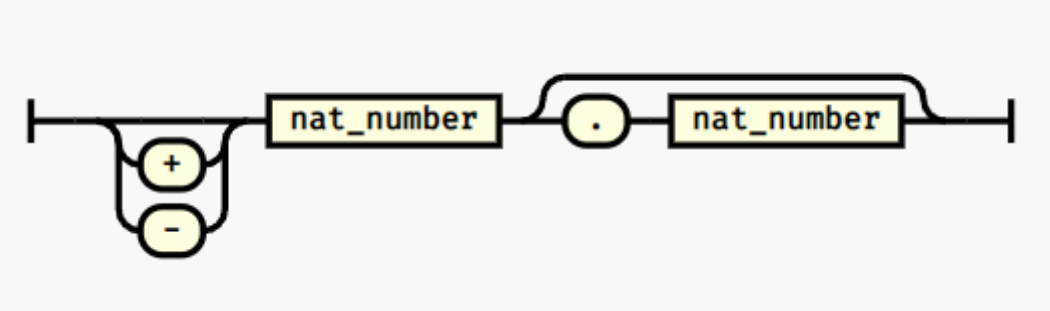
Натуральные числа

```
nat_number = digit, { digit };
```



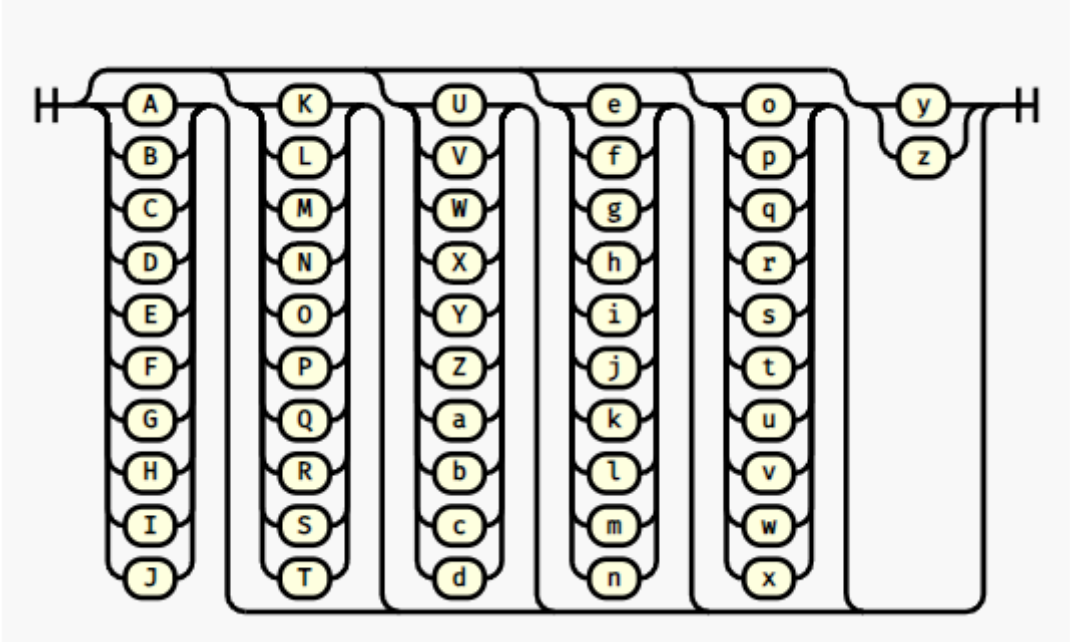
Числа

```
number = ["+" | "-"], nat_number, [ ".", nat_number ];
```



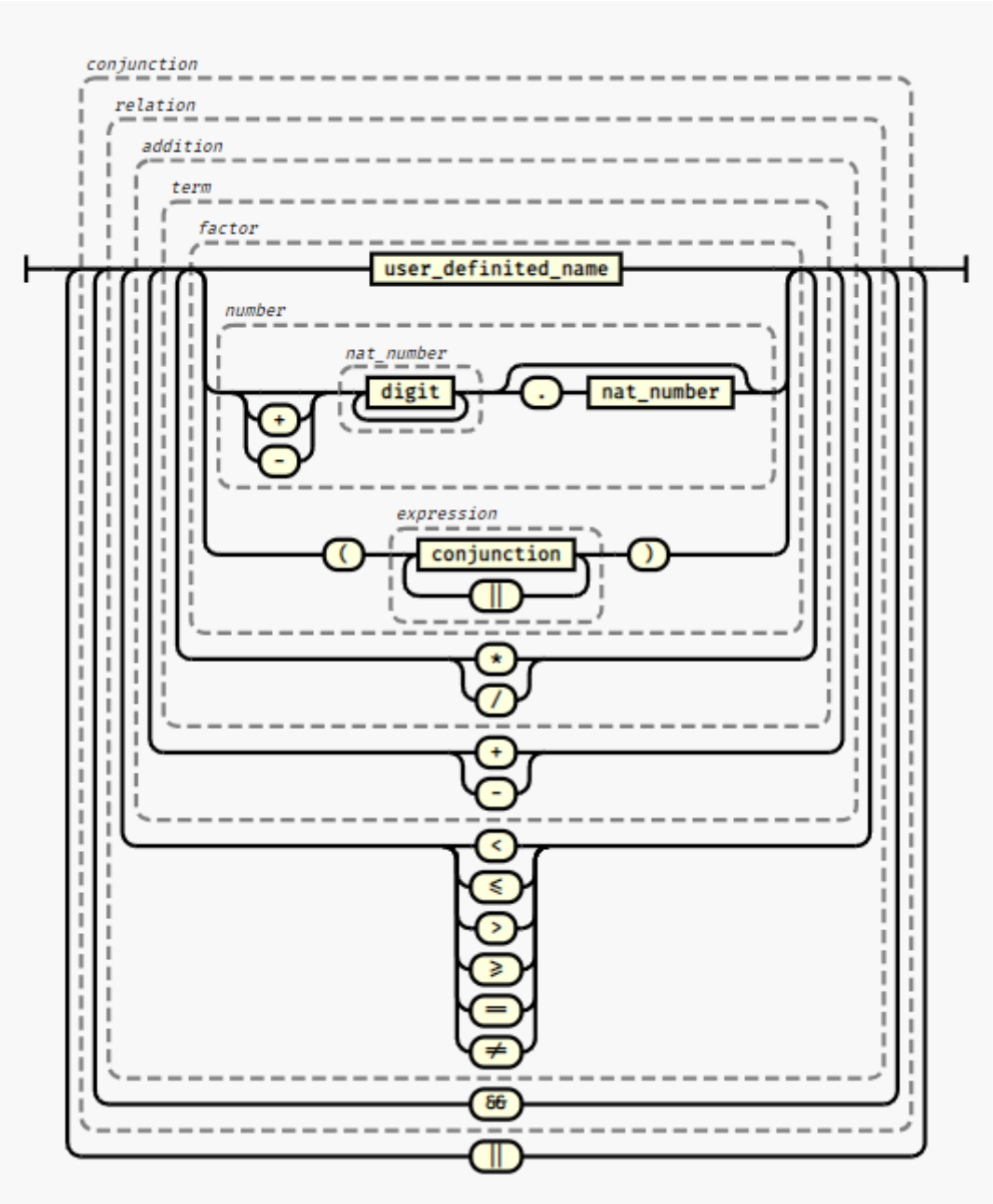
Буквы

```
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" ;
```



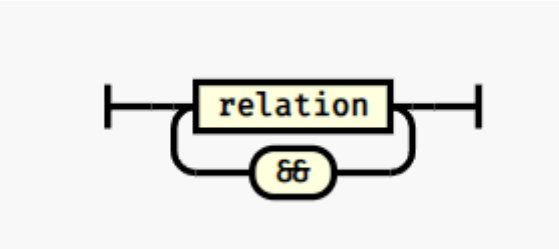
Выражение

```
expression = conjunction, { "||", conjunction };
```



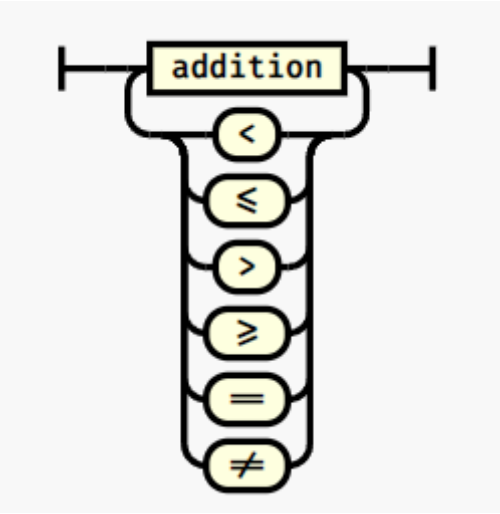
Коньюнкция

```
conjunction = relation, { "&&", relation };
```



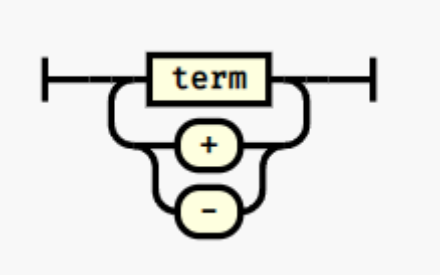
Дизьюнкция

```
relation = addition, { ("<" | "⩽" | ">" | "⩾" | "=" | "≠"), addition
};
```



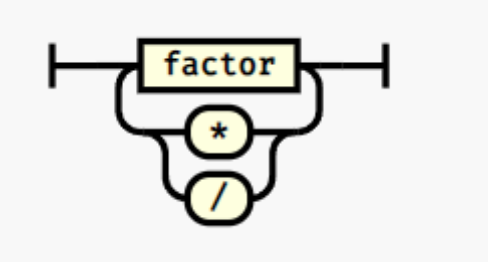
Сложение и вычитание

```
addition = term, { ("+" | "-"), term };
```



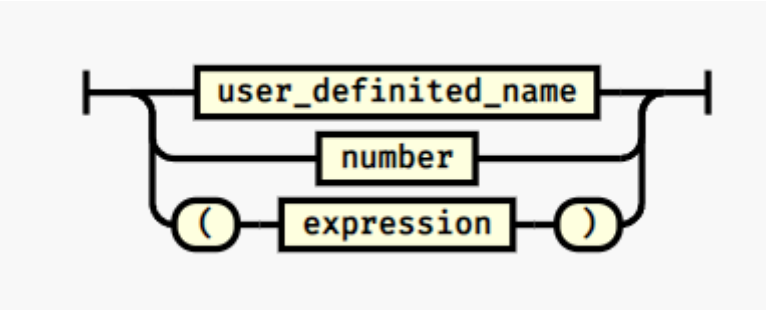
Умножение и деление

```
term = factor, { ("*" | "/"), factor };
```



Компоненты операций

```
factor = user_definited_name | number | "(", expression, ")";
```


user_definited_name
number
(
expression
)