

Федеральное государственное автономное образовательное учреждение
высшего образования
Национальный исследовательский университет
«Высшая школа экономики»

Факультет социально-экономических и компьютерных наук

Лабораторная работа №3

по дисциплине «Теоретические основы информатики»

Раздел 5 – Языки программирования и основы трансляции

Выполнил: студент Яцишин Л.С., уч. группа ПСАПР-25-2

Пермь, 2025

Содержание

1	Этап 1. Разработка расширенной грамматики выражений	3
	Этап 1 – Алфавит и лексемы	3
	Этап 1 - Грамматика	3
	Этап 1 – Диаграммы Вирта	4
	Этап 1 - Примеры корректных выражений	8
	Этап 1 – Примеры выражений с ошибками	8
	Этап 1 – Покрытие правил тестами	8
	Этап 1 - Пример вывода и дерево разбора	9

1 Этап 1. Разработка расширенной грамматики выражений

Разработать расширенную грамматику выражений (включая правила записи операндов – чисел и переменных), представить её в форме РБНФ и диаграмм Вирта. Привести примеры корректных и ошибочных выражений, построить выводы и/или деревья синтаксического разбора хотя бы для одного выражения. Допускаются разделители (пробелы) между лексемами, при разборе они пропускаются.

Алфавит и лексемы

Алфавит. Используются цифры, латинские буквы (для идентификаторов), знаки арифметических операций, сравнения, скобки и разделители аргументов функций:

$$A = \{ '0', '1', '2', '3', '4', '5', '6', '7', '8', '9', \\ 'a'..'z', 'A'..'Z', '+', '-', '*', '/', \\ '(', ')', '=', '<', '>', ',', '.', ',' \}$$

Ключевые слова задаются как отдельные терминальные лексемы: IF, AND, OR, div, mod. В грамматике они рассматриваются как отдельные терминалы (уровень токенов). Пробелы и табуляция не входят в алфавит и при разборе просто пропускаются.

Лексемы. Идентификаторы – буквы и цифры, начинаются с буквы. Числа – целые и вещественные (с точкой или запятой), знак задаётся на уровне унарного оператора:

- целые: 7, 00, 789;
- вещественные: 7., 57.00, .9, 0.015.

Ключевые слова IF, AND, OR, div, mod запрещены как $\langle \text{Id} \rangle$.

Грамматика

Расширенная LL(1)-грамматика, построенная пошаговой детализацией.

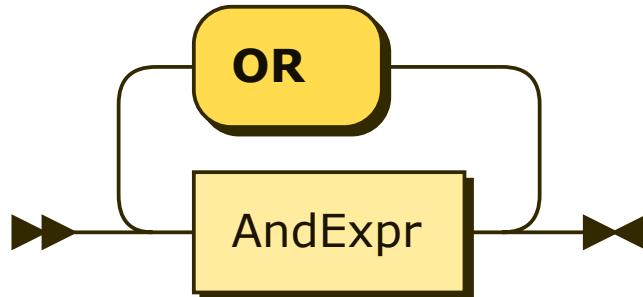
```
<Expression> ::= <OrExpr>
<OrExpr> ::= <AndExpr> { 'OR' <AndExpr> }
<AndExpr> ::= <CmpExpr> { 'AND' <CmpExpr> }
<CmpExpr> ::= <AddExpr> [ <CmpOp> <AddExpr> ]
<CmpOp> ::= '<' | '<=' | '=' | '><' | '>=' | '>'
<AddExpr> ::= <MulExpr> { ('+' | '-') <MulExpr> }
<MulExpr> ::= <Unary> { '*' | '/' | 'div' | 'mod' } <Unary>
<Unary> ::= [ '+' | '-' ] <Primary>
<Primary> ::= <Number> | <Id> | '(' <Expression> ')' | <IfExpr>
<IfExpr> ::= 'IF' '(' <Expression> ';' <Expression> ';' <Expression> ')'
<Number> ::= <UnsignedInt> [ ('.' | ',') [<UnsignedInt>] ] | ('.' | ',') <UnsignedInt>
<UnsignedInt> ::= <Digit> { <Digit> }
<Id> ::= <Letter> { <Letter> | <Digit> }
<Digit> ::= '0' | '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9'
<Letter> ::= 'a'..'z' | 'A'..'Z'
```

Диаграммы Вирта

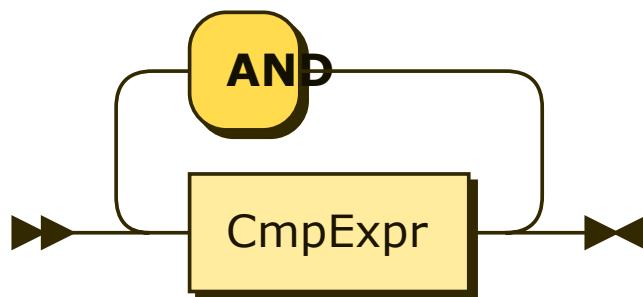
<Expression>



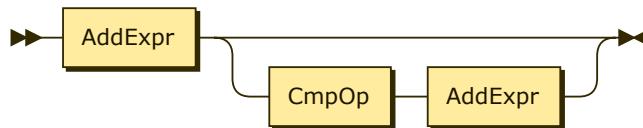
<OrExpr>



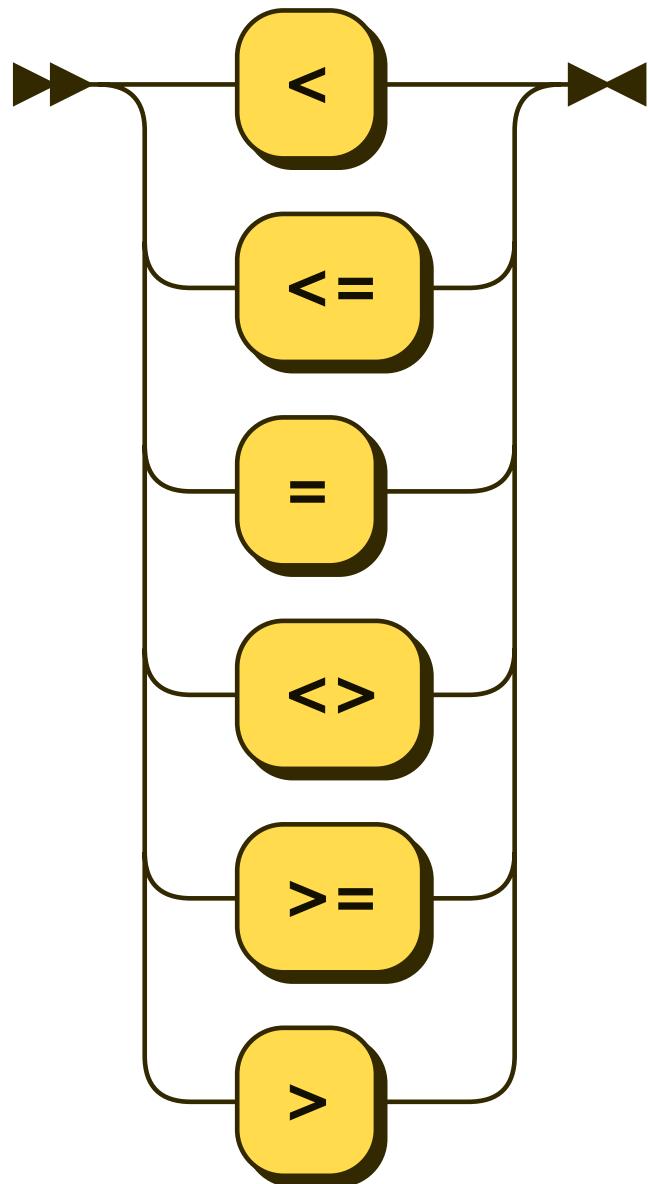
<AndExpr>



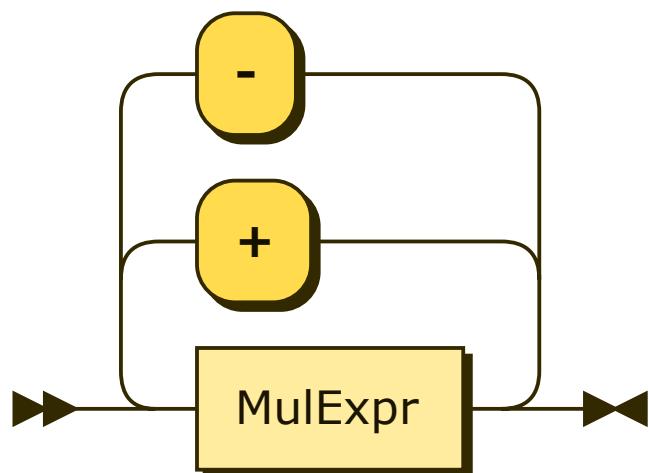
<CmpExpr>



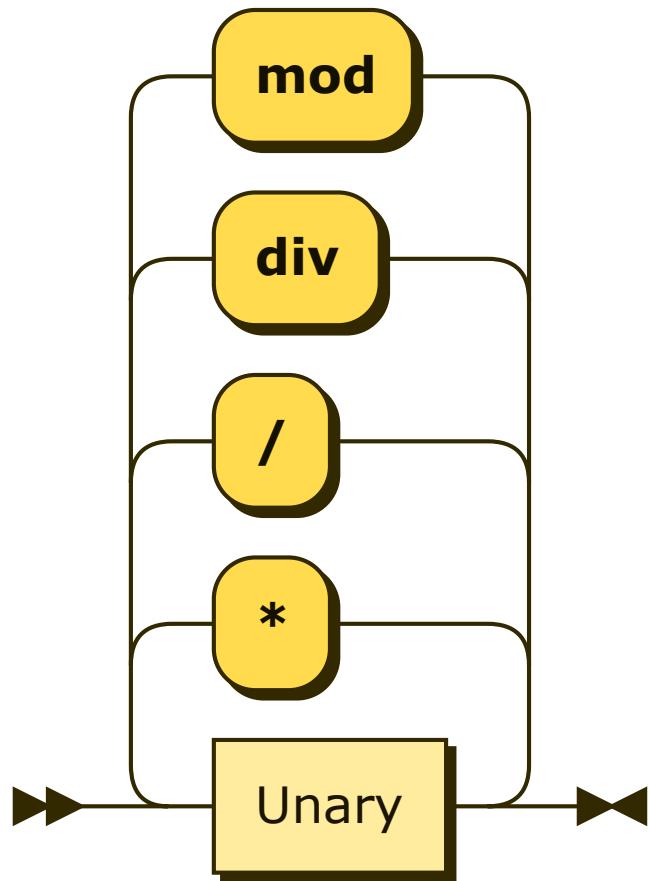
<CmpOp>



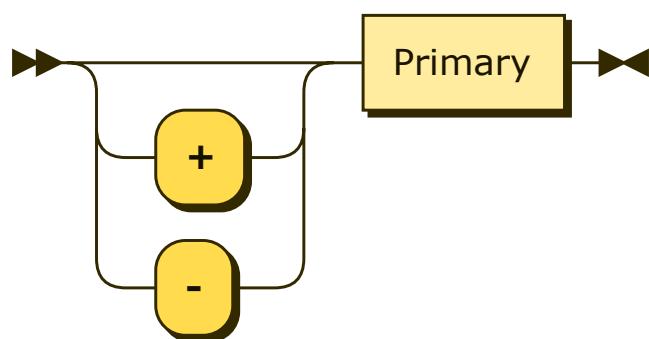
<AddExpr>



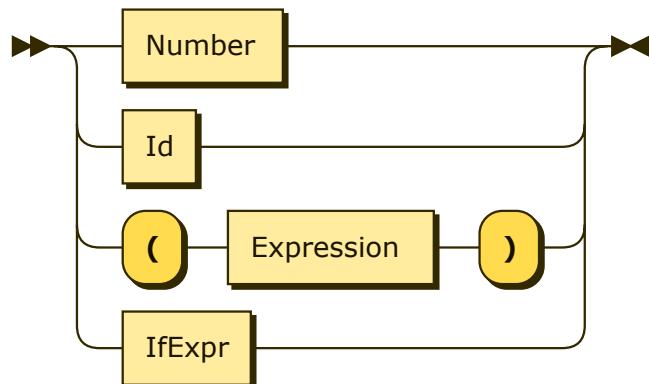
<MulExpr>



<Unary>



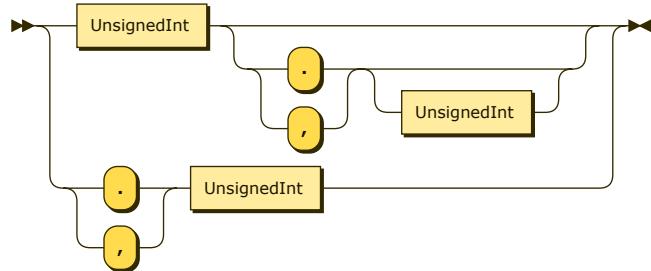
<Primary>



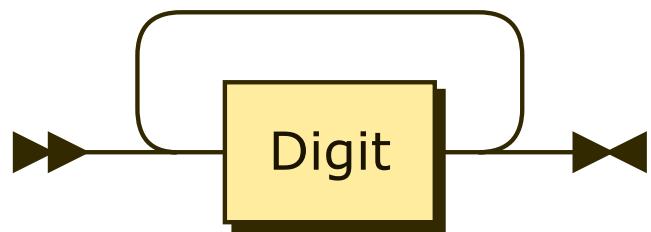
<IfExpr>



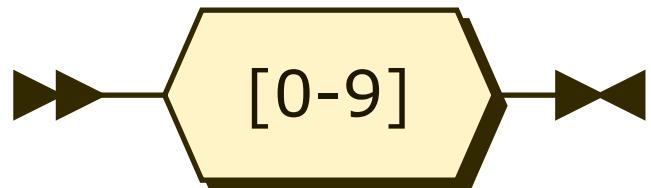
<Number>



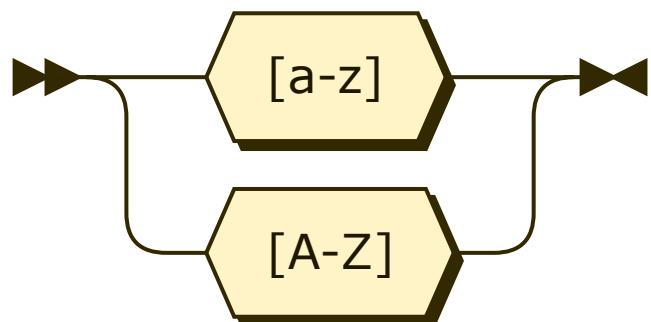
<UnsignedInt>



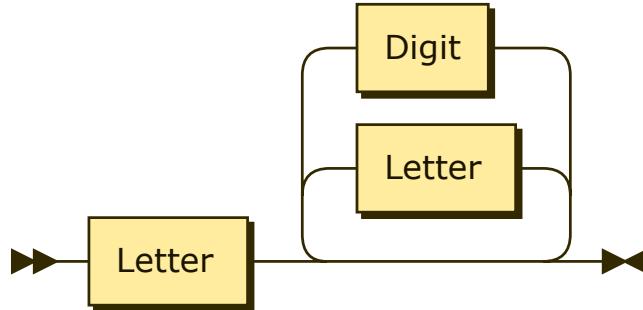
<Digit>



<Letter>



<Id>



Примеры корректных выражений

- A1 $a+5$
- A2 $-12.5 + b*7$
- A3 $(100-f)/d + 8*(a-99) - a*b/100*7$
- A4 IF($a < 5$; 10; 10-a)
- A5 $(a > 0 \text{ AND } b > 0) \text{ OR } c = 0$
- A6 $x \text{ div } 3 + y \text{ mod } 2$
- A7 $.9 + 7. + +00$
- A8 $0,5 + 1$
- A9 $a \leq b$
- A10 $a \neq 0$

Примеры выражений с ошибками

- B1 $+ 2 3$ – после оператора ожидается один operand; затем встретился ещё один operand без операции.
- B2 $a+*3$ – после + ожидается множитель.
- B3 IF($a < 5; 10$) – отсутствуют второй разделитель ; и третий аргумент.
- B4 $s-(75/k+r)/a+$ – выражение оканчивается знаком операции.
- B5 $(10-5 -$ нет закрывающей скобки.

Покрытие правил тестами

Правило / конструкция	Примеры
Числа: целые/вещественные, разные формы	A2, A7, A8
Идентификаторы и операции +, -, *, /	A1, A2, A3
Скобки и приоритеты (вложения)	A3, пример вывода $25+A^*(10-A)$
Сравнения (<, >, =, <=, <>)	A4, A5, A9, A10
Логика AND/OR	A5
Целочисленные операции div/mod	A6
Условное выражение IF(...)	A4
Ошибки: два операнда подряд / пропуск операции	B1
Ошибки: неверное начало множителя	B2
Ошибки: неполный вызов IF	B3
Ошибки: выражение оканчивается оператором	B4
Ошибки: нет закрывающей скобки	B5

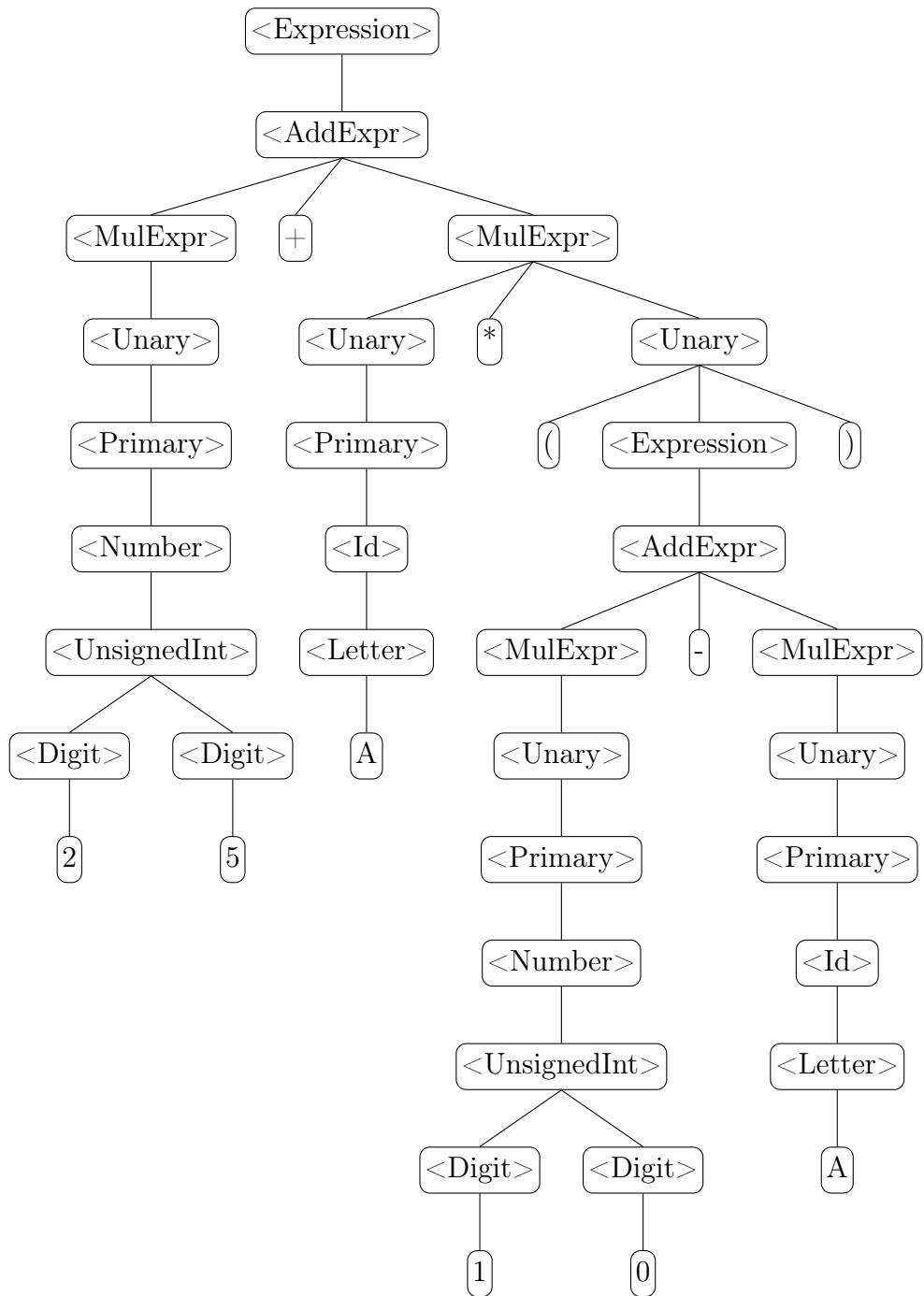
Пример вывода и дерево синтаксического разбора

Рассмотрим выражение: $25+A^*(10-A)$.

Цепочка вывода (фрагмент).

$$\begin{aligned}
 \langle Expression \rangle &\Rightarrow \langle OrExpr \rangle \\
 &\Rightarrow \langle AndExpr \rangle \\
 &\Rightarrow \langle CmpExpr \rangle \\
 &\Rightarrow \langle AddExpr \rangle \\
 &\Rightarrow \langle MulExpr \rangle + \langle MulExpr \rangle \\
 &\Rightarrow \langle Number \rangle + \langle MulExpr \rangle \\
 &\Rightarrow 25 + \langle Unary \rangle * \langle Unary \rangle \\
 &\Rightarrow 25 + A * (\langle Expression \rangle) \\
 &\Rightarrow 25 + A * (10 - A)
 \end{aligned}$$

Дерево разбора.



Дополнительные фрагменты выводов (расширения).

$$\begin{aligned}
 <\text{Expression}> &\Rightarrow <\text{OrExpr}> \Rightarrow <\text{AndExpr}> \text{ OR } <\text{AndExpr}> \\
 &\Rightarrow (<\text{CmpExpr}> \text{ AND } <\text{CmpExpr}>) \text{ OR } <\text{CmpExpr}> \\
 &\Rightarrow (a > 0 \text{ AND } b > 0) \text{ OR } c = 0
 \end{aligned}$$

$$\begin{aligned}
 \langle Expression \rangle &\Rightarrow \langle OrExpr \rangle \\
 &\Rightarrow \langle AndExpr \rangle \\
 &\Rightarrow \langle CmpExpr \rangle \\
 &\Rightarrow \langle AddExpr \rangle \\
 &\Rightarrow \langle MulExpr \rangle \\
 &\Rightarrow \langle Unary \rangle \\
 &\Rightarrow \langle Primary \rangle \\
 &\Rightarrow \langle IfExpr \rangle \\
 &\Rightarrow \text{IF}(a < 5; 10; 10 - a)
 \end{aligned}$$

Пояснение по локализации ошибок. Если при разборе встречен символ, который не подходит ни под одну альтернативу в текущем правиле (например, после знака + встретился *), разбор останавливается. Сообщение содержит позицию символа, ожидаемую конструкцию и причину ошибки (неверное начало операнда, отсутствие закрывающей скобки, конец строки и т. д.).