Министерство образования Республики Беларусь

Учреждение образования

Белорусский государственный университет информатики и

Радиоэлектроники

Факультет информационных технологий и управления

Кафедра интеллектуальных информационных технологий

Отчет по лабораторной работе №1

по курсу “Логические основы интеллектуальных систем”

Вариант 3

Выполнил:

Студент гр. 221701 Крупский А. В.

Проверил: Ивашенко В. П.

Минск

2024

**Тема:**

Представление и синтаксическая проверка формул языка логики высказываний.

**Цель:**

Приобрести навыки программирования алгоритмов синтаксического разбора формул языка логики высказываний.

**Задание:**

Проверить, является ли формула сокращенного языка логики высказываний невыполнимой (противоречивой).

**Дополнительно:**

Предусмотреть работу системы в режиме тестирования знаний пользователя.

**Грамматика языка логики высказываний:**

<константа> ::=1|0

<символ> ::=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

<отрицание> ::= !

<конъюнкция> ::= /\

<дизъюнкция> ::= \/

<импликация> ::= ->

<эквиваленция> ::= ~

<открывающая скобка> ::= (

<закрывающая скобка> ::= )

<бинарная связка> ::= <конъюнкция>|<дизъюнкция>|<импликация>|<эквиваленция>

<атомарная формула> ::= <латинская заглавная буква>

<унарная сложная формула> ::= <открывающая скобка><отрицание><формула><закрывающая скобка>

<бинарная сложная формула> ::= <открывающая скобка><формула><бинарная связка><формула><закрывающая скобка>

<формула> ::=<логическая константа> |<атомарная формула>|<сложная формула>

Невыполнимая формула -- формула, ложная на всех её интерпретациях.

**Блок-схемы функций программы:**

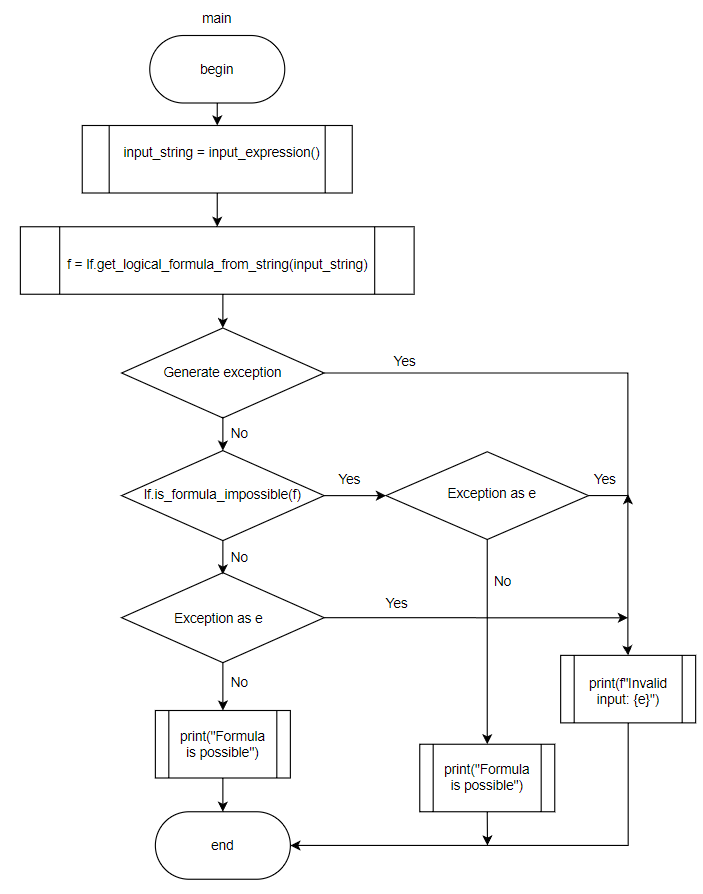


Рис 1. Функция main().

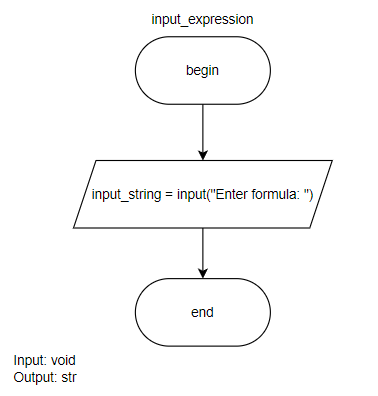


Рис 2. Функция input\_expression().

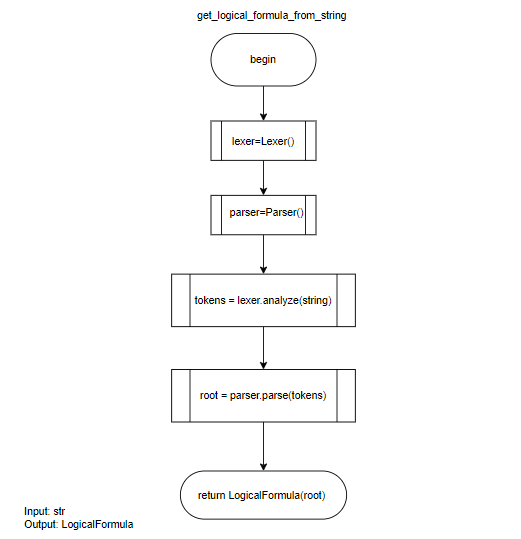
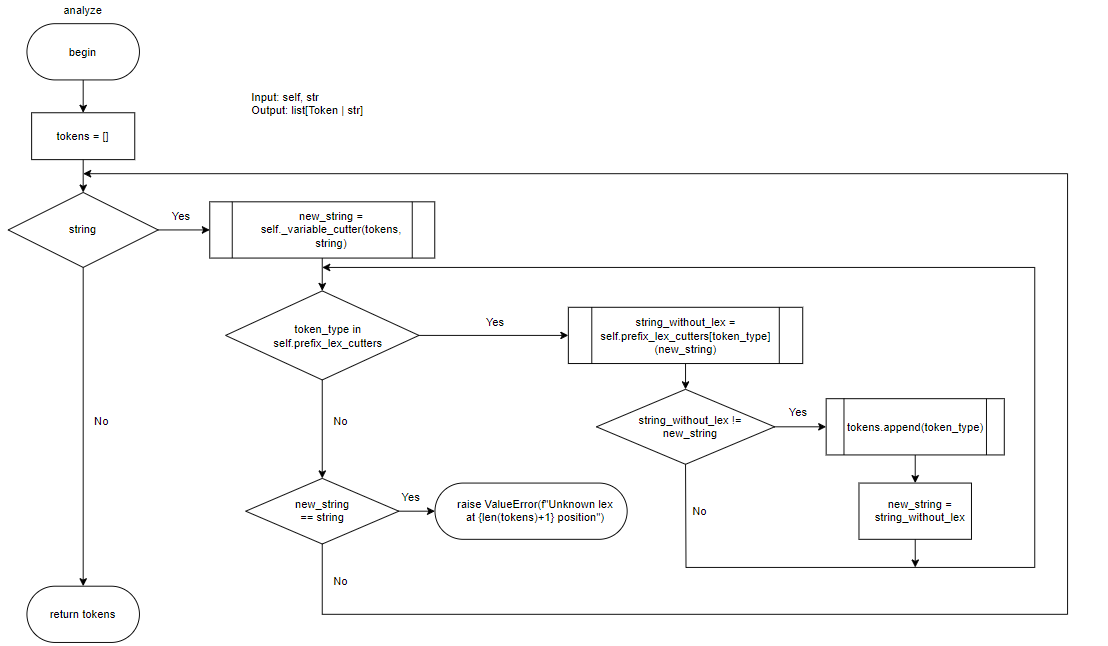


Рис 3. Функция get\_logical\_formula\_from\_string().

Рис 4. Метод analyze() класса LogicalFormulaLexer.

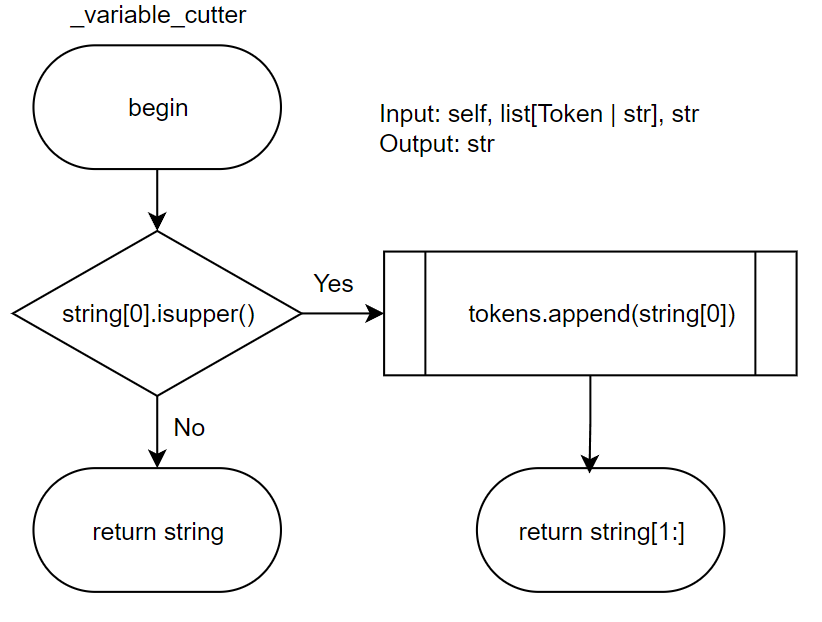


Рис 5. Метод \_variable\_cutter() класса LogicalFormulaLexer.

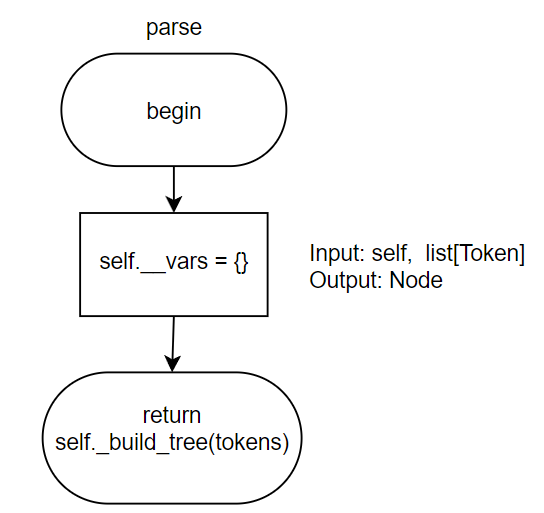


Рис 6. Метод parse() класса LogicalFormulaParser.

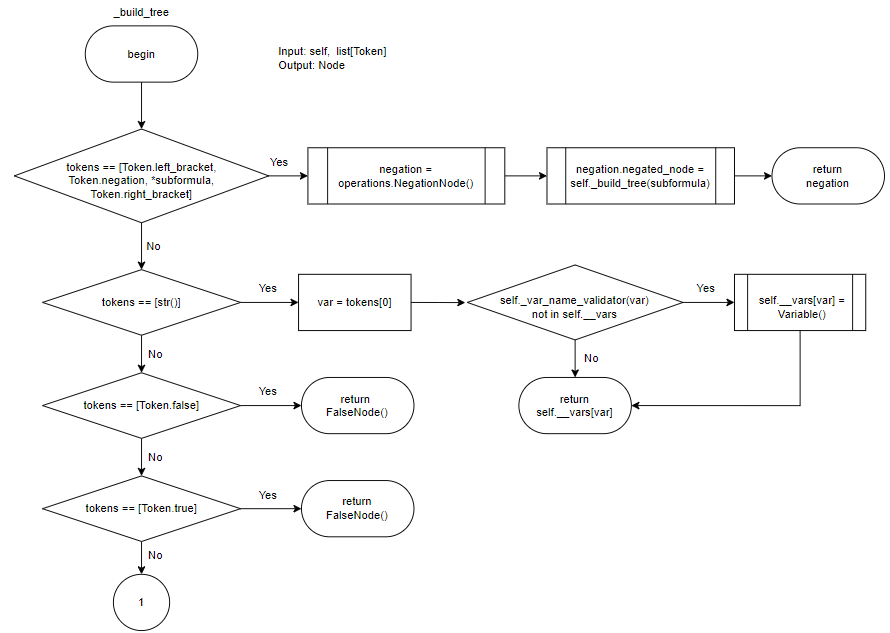


Рис 7-1. Метод \_build\_tree() класса LogicalFormulaParser.

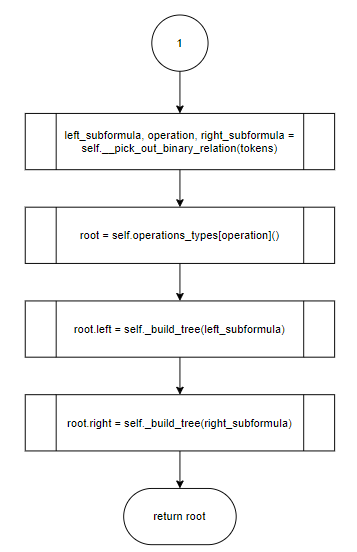


Рис 7-2. Метод build\_tree() класса LogicalFormulaParser.

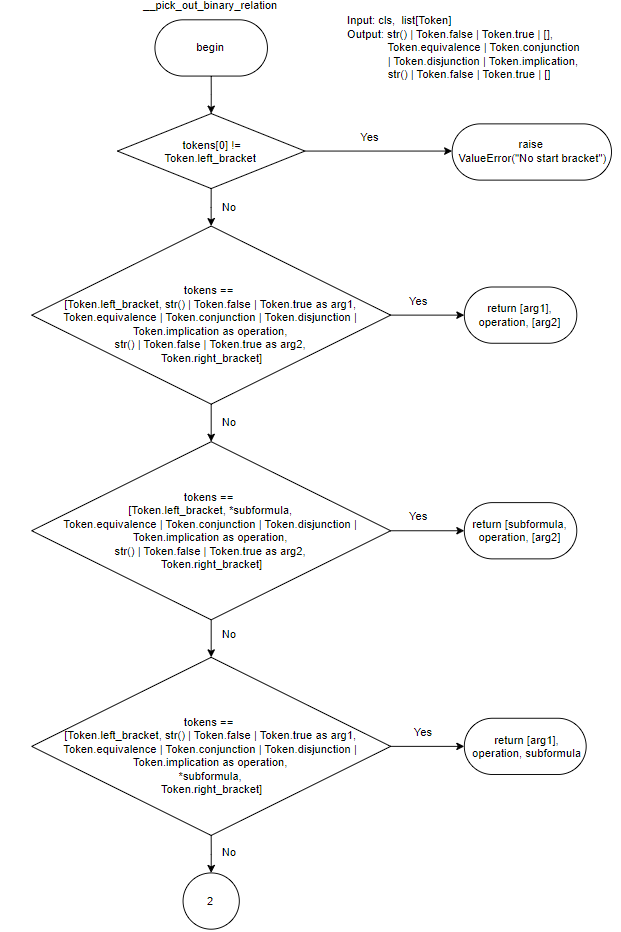


Рис 8-1. Метод \_\_pick\_out\_binary\_relation() класса LogicalFormulaParser.

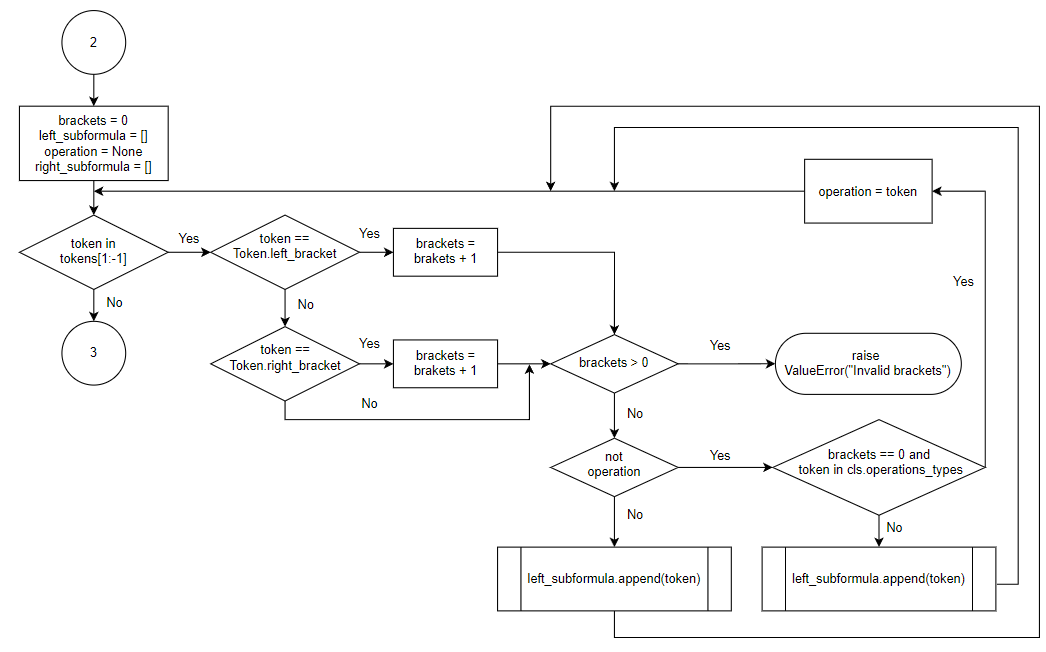


Рис 8-2. Метод \_\_pick\_out\_binary\_relation() класса LogicalFormulaParser.

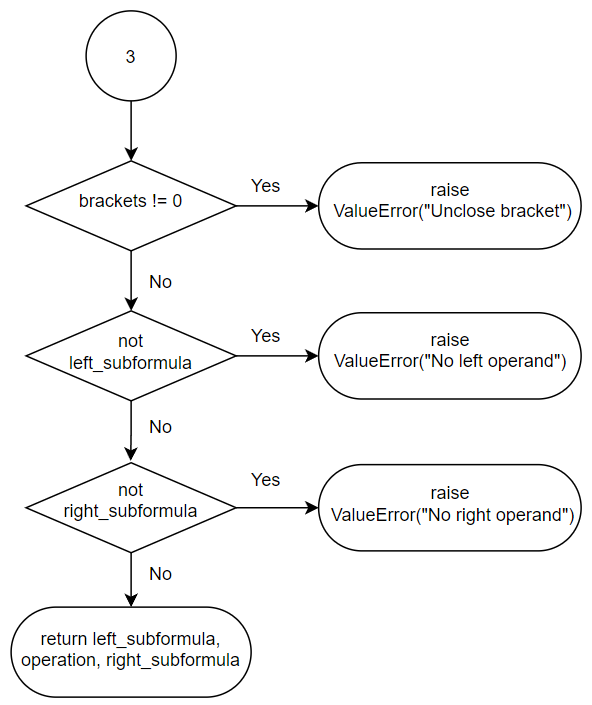


Рис 8-3. Метод \_\_pick\_out\_binary\_relation() класса LogicalFormulaParser.

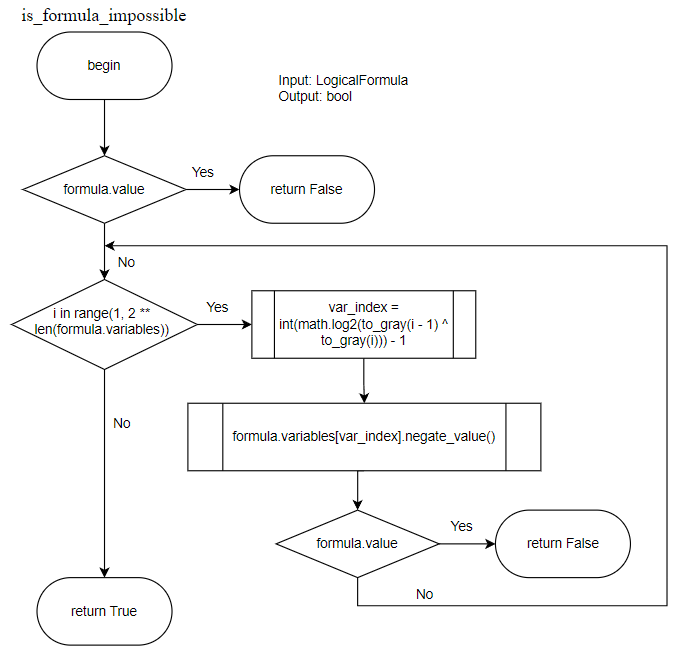


Рис 9. Функция is\_formula\_impossible()

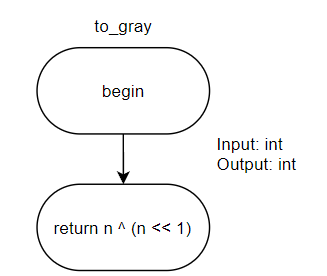


Рис 10. Функция to\_gray().

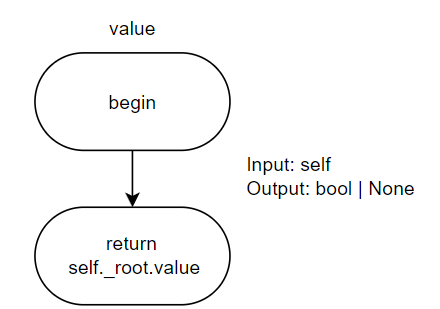


Рис 11. Метод-геттер value() класса LogicalFormula.

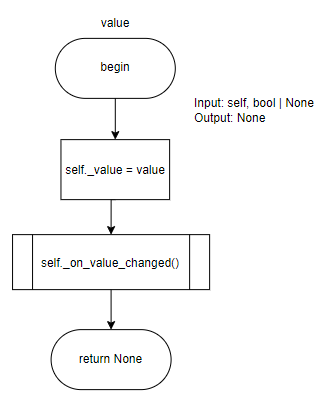


Рис 12. Метод-сеттер value() класса VariableNode.

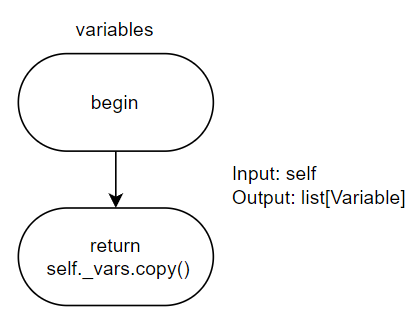


Рис 13. Метод-геттер variables() класса LogicalFormula.

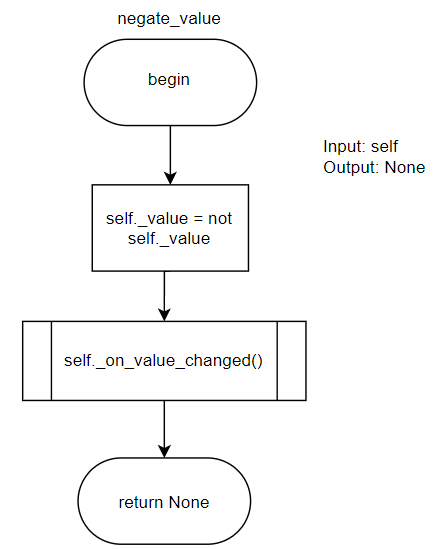


Рис 14. Метод negate\_value() класса VariableNode.

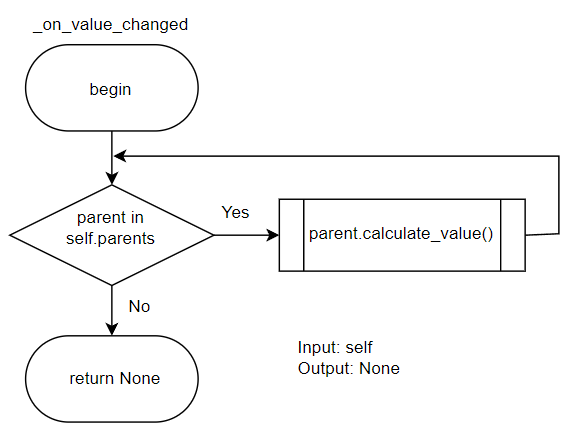


Рис 15. Метод \_on\_value\_changed() класса NonConstantNode.

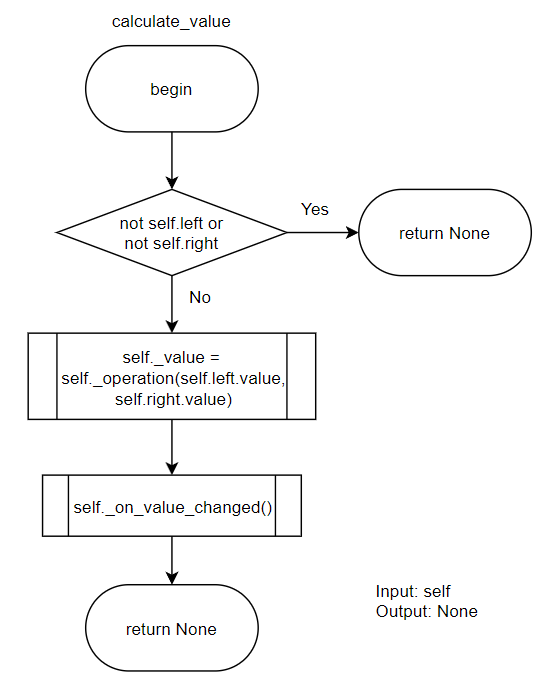


Рис 16. Метод calculate\_value() класса BinaryOperation.

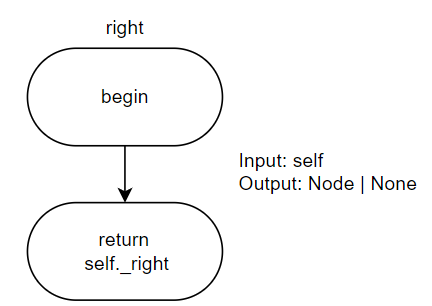


Рис 17. Метод-геттер right() класса BinaryOperation.

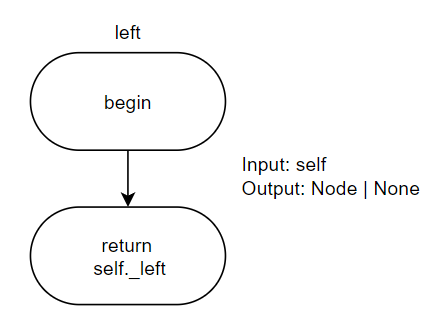


Рис 18. Метод-геттер left() класса BinaryOperation.

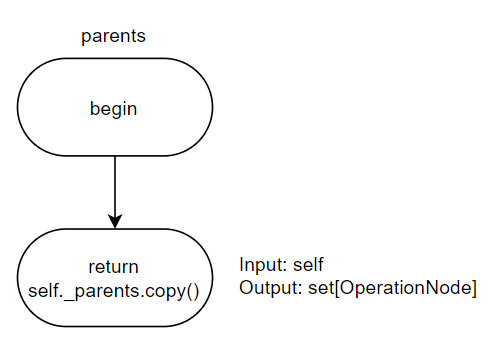


Рис 19. Метод-геттер parents() класса NonConstantNode.

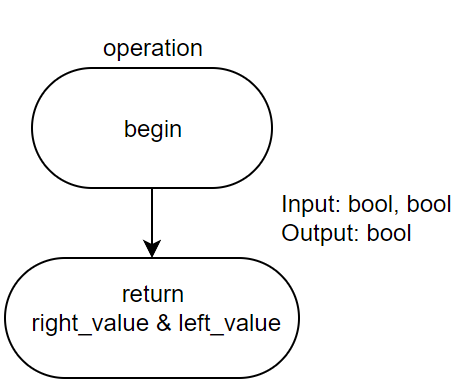


Рис 20. Метод operation() класса ConjunctionNode.

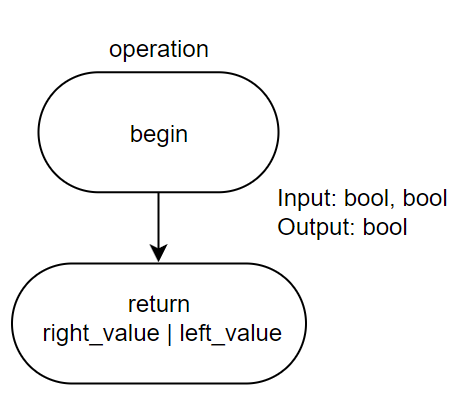


Рис 21. Метод operation() класса DisjunctionNode.

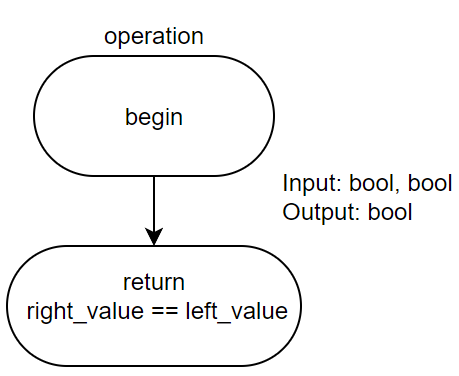


Рис 22. Метод operation() класса EquivalenceNode.

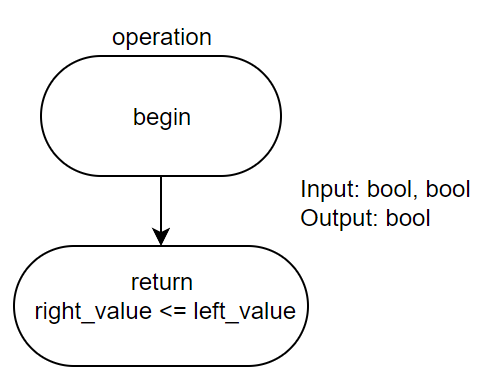
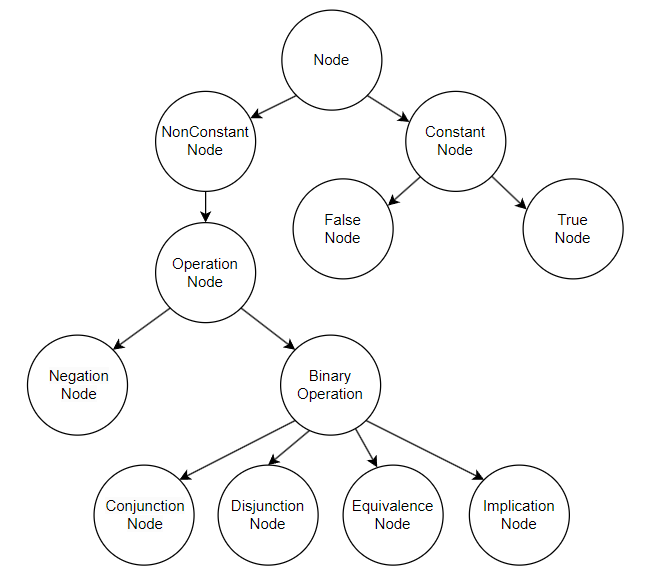


Рис 23. Метод operation() класса ImplicationNode.

**Иерархия классов узлов:**

**Выполнение:**

Данная лабораторная работа выполнялась в бригаде, состоящей из двух человек: Глёза Е.Д. (221701) и Крупский А.В. (221701).

Для выполнения данной лабораторной работы был использован язык программирования Python. Основой кода является синтаксическое бинарное дерево, где каждый узел представляет собой логическую связку. В методах использовались структуры данных стандартной библиотеки: список, строка, словарь.

После запуска программы программа запрашивает пользователя ввести логическую формулу.



Рис 24. Ввод формулы.

Если пользователь ввел некорректную формулу, то программа выведет сообщение об ошибке, после которого происходит завершение программы:

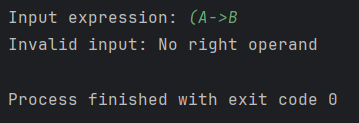


Рис 25. Вывод сообщения об ошибке.

Если пользователь ввел корректную формулу, то программа выведет сообщение, является ли логическая формула выполнимой или нет, после которого происходит завершение программы:

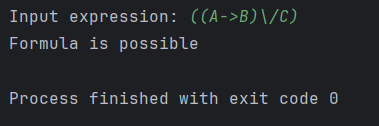


Рис 26. Вывод сообщения результата проверки.

**Примеры:**

Рис 27. Пример 1

Рис 28. Пример 2

Рис 29. Пример 3

Рис 30. Пример 4

**Пример выполнения тестовых заданий:**

Рис 31. Пример 1

Рис 32. Пример 2

**Вывод:**

В ходе работы были приобретены навыки программирования алгоритмов синтаксического разбора формул языка логики высказываний.

Также была реализована процедурная программа, синтаксического разбора формул языка логики высказываний, проверяющая, является ли формула сокращенного языка логики высказываний невыполнимой (противоречивой).

**Теоретические сведения были взяты из следующих источников:**

1. Логические основы интеллектуальных систем. Практикум: учебно-методическое пособие / В. В. Голенков, В. П. Ивашенко, Д. Г. Колб, К. А. Уваров. – Минск: БГУИР, 2011.