Министерство образования Республики Беларусь

Учреждение образования

“Белорусский государственный университет

информатики и радиоэлектроники”

Факультет информационных технологий и управления

Кафедра интеллектуальных информационных технологий

**Лабораторная работа №1**

по дисциплине «Логические основы интеллектуальных систем»

Вариант №B

Выполнил: Name

Группа: Group

Проверил: Name

Минск Year

**Содержание**

1. Постановка задачи 2

1.1. Задание 2

1.2. Входные данные 2

1.3. Выходные данные 2

2. Описание реализации 2

2.1. Структура данных 2

2.2. Алгоритм 3

2.3. Блок схема алгоритма 4

2.4. Описания тестирования алгоритма 7

3. Вывод 8

# **1. Постановка задачи**

## **1.1. Задание**

Проверить является ли строка формулой сокращенного языка логики высказывания.

## **1.2. Входные данные**

На вход программе подаётся строка. Программа проверяет записана ли данная строка по грамматика сокращённого языка логических выражений.

## **1.3. Выходные данные**

Логическое значение «Да» (True) или «Нет» (False), которое отражает то, соответствует ли переданная строка грамматике сокращённого языка логических выражений.

# **2. Описание реализации**

## **2.1. Структура данных**

Для хранения структуры логической формулы используется граф. Данные каждая вершины включают из индекс, тип формулы и формулу. Ребрами соединяются вершины формулы и подформула. Чаще всего данный граф выглядит как несбалансированное бинарное дерево.

Пример: ((P&(Q|R))->((P&Q)|R))

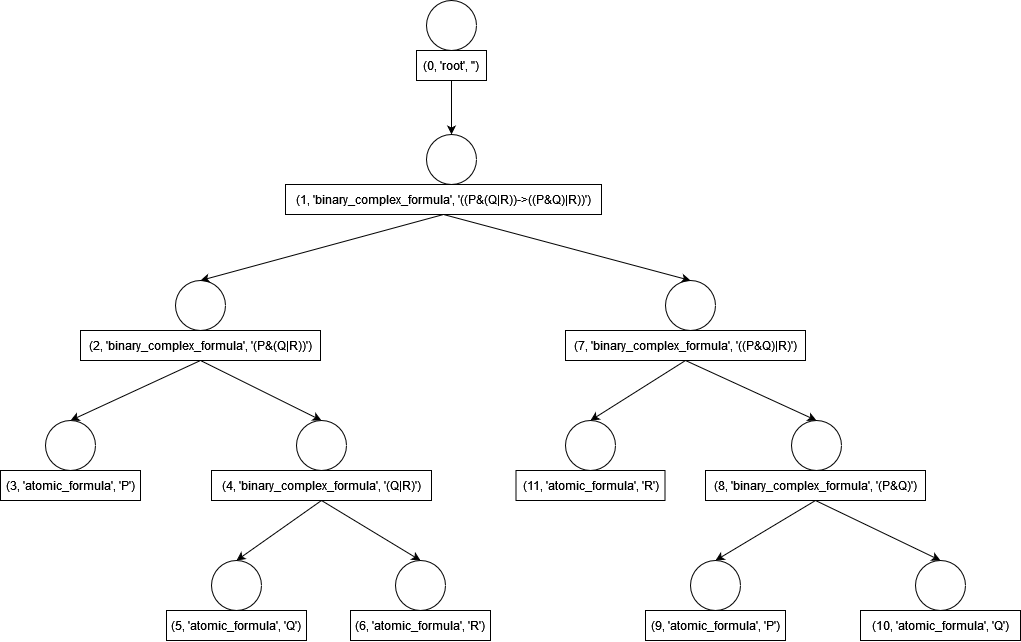


Рисунок 1. Пример схема структуры хранения формулы в программе.

В методах использовались структуры данных стандартной библиотеки: Списки и кортежи (с реализацией на основе массива), Ассоциативный массив (с реализацией на основе массива и хеш-функции), строки.

## **2.2. Алгоритм**

Основной алгоритм начинает свою работу при передаче введенной пользователем строки в качестве аргумента в рекурентный метод LogicFormulaParser.is\_formula():

Разбор строки происходит следующим образом:

1. Разбор текста логического выражения опираясь на вложенность функций
   1. Проверка синтаксиса атомарной константы
   2. Проверка синтаксиса атомарной формулы
   3. Проверка синтаксиса унарной сложной формулы
   4. Проверка синтаксиса бинарной сложной формулы

## **2.3. Блок схема алгоритма**

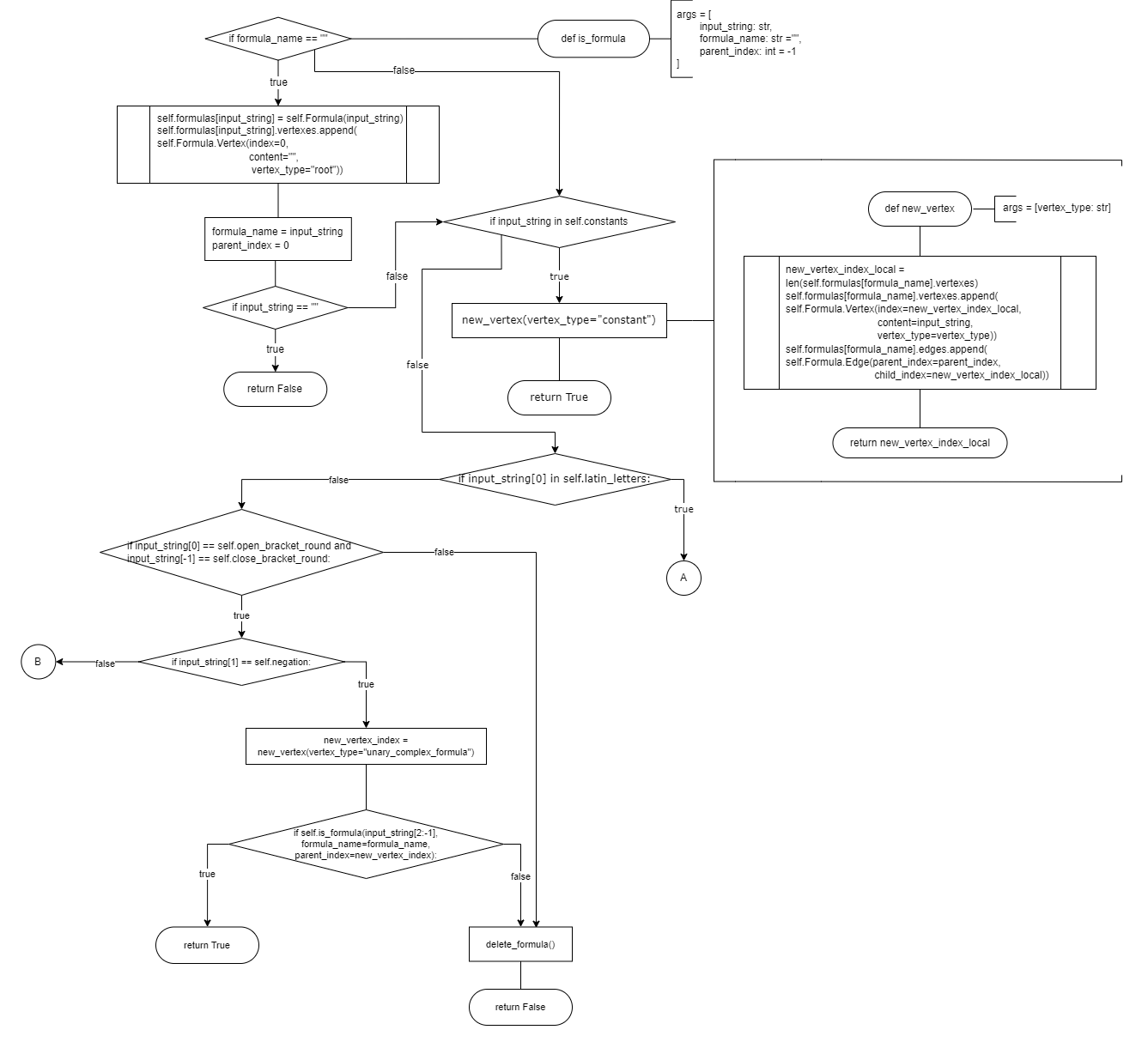
****

Рисунок 2. Схема алгоритма часть 1.

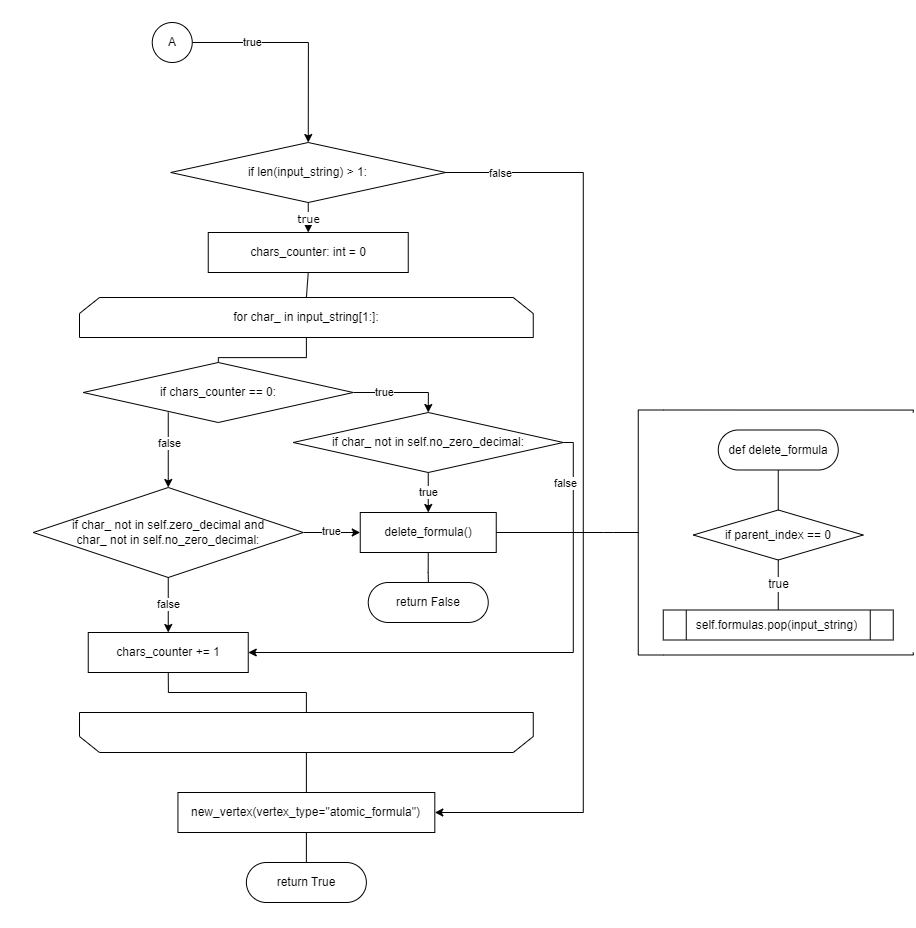


Рисунок 3. Схема алгоритма часть 2.

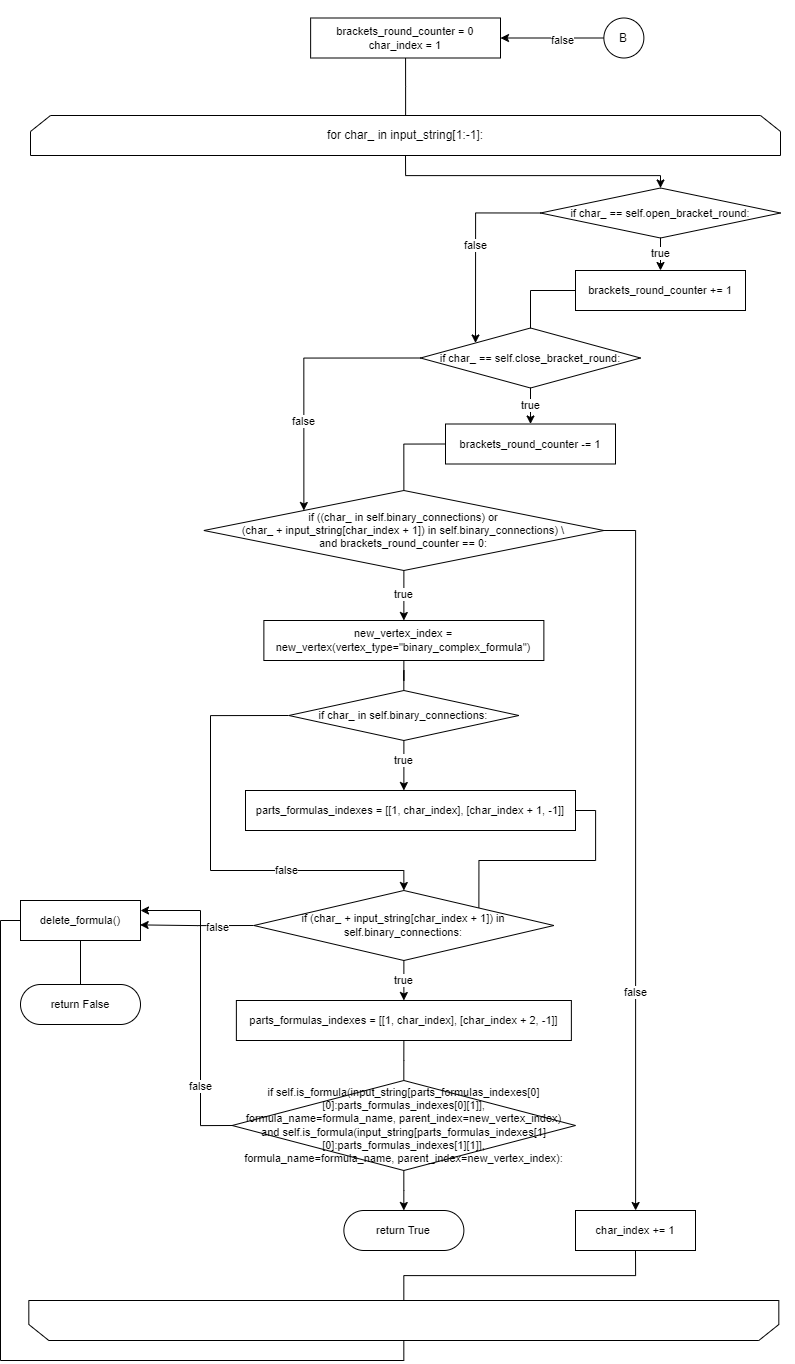


Рисунок 4. Схема алгоритма часть 3.

## **2.4. Описания тестирования алгоритма**

В программе присутствует система тестирования. Для ее использования можно либо вызвать отдельно метод тестирования из экземпляра класса парсера, либо при создания данного экземпляра, указать аргумент для “test=True”. Система последовательно выбирает строку из файла tests.csv, передает данную строку в метод с алгоритмом, и сравнивает ответ программы и сохраненный в файле ответ.

pars = LogicFormulaParser(test=True)

pars.\_\_test\_parser\_\_()

# **Вывод**

В данной лабораторной работе был реализован парсер формулы, проверяющий заданную строку на соответствие языку логических высказываний заданной грамматики.