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»

на тему

«Представление и синтаксическая проверка формул языка логики высказываний»

Вариант F

Выполнил студент гр. 121702 Заломов Р.А.

Проверил Ивашенко В. П.

Минск 2023

Цель: приобрести навыки алгоритмизации синтаксического разбора формул языка логики высказываний.

Задача: проверить, является ли данная формула ДНФ.

Описание лабораторной работы

По ходу лабораторной работы необходимо проверить, является ли введённая формула ДНФ. Для этого необходимо:

1. Проверить, принадлежат ли все символы формулы алфавиту языка логики высказываний.
2. Проверить, соответствует ли данная формула правилам синтаксиса языка логики высказываний.
3. Проверить, соответствует ли формула правилам формул, имеющих вид ДНФ.

Для этих задач были синтезированы следующие подзадачи

1. Анализ символов, входящих в строку, содержащую формулу
2. Проверка правильности расстановки скобок в формуле
3. Проверка формулы на соответствия правилам формул в дизъюнктивной нормальной форме:
    3.1. Проверить формулу на наличие отрицания только атомарных формул (без констант)
    3.2. Проверить последовательность логических операций в формуле на правильность. Под правильностью понимается соответствие характера последовательности таковой для формул языка логики высказываний в ДНФ.

Теоретические сведения

Алфавит языка логики высказываний — алфавит, включающий символы логических констант и логических связок, символы для обозначения высказываний, скобки для указания приоритета операций (45 символов: 2 логических константы, десятичные цифры, заглавные буквы латинского алфавита для обозначения высказываний, 5 логических связок).

Алфавит – конечное или счетное множество символов.

Множество — абстрактная сущность, непосредственно связывающая одну или несколько сущностей в целое.

Абстрактный — существующий во внутренней памяти субъекта.

Субъект — носитель действия.

Действие — явление, которое имеет событие, предшествующее всем остальным событиям.

Целое — отнесенное к себе или к своим частям.

Отношение — множество связок.

Связка — абстрактная связь, множество не менее чем из одного элемента.

Формальный язык — множество текстов формального языка над некоторым алфавитом.

Грамматика формального языка состоит из правил вида п::=ф.

Грамматика языка логики высказываний:

<логическая константа> ::= 1|0

<латинская заглавная буква> ::= 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

<формула> ::= <логическая константа> | < латинская заглавная буква > | <унарная сложная формула> | <бинарная сложная формула>

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

<открывающая скобка> ::= (

<отрицание> ::= !

<закрывающая скобка> ::=)

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

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

<конъюнкция> ::= /\

<дизъюнкция> ::= \/

<импликация> :: = ->

<эквиваленция> ::= ~

Подформула языка логики высказываний — формула языка логики высказываний, которая является подстрокой формулы языка логики высказываний.

Литерал – атомарная формула (без констант) или её логическое отрицание.

Дизъюнктивная нормальная форма (ДНФ) – нормальная форма, в которой формула языка логики высказываний имеет вид дизъюнкции конъюнкций литералов.

Примеры формул в ДНФ:

(A\/B)

(A/\B)

((A/\B)\/(!A))

((C/\B)\/(D/\E))

(((A/\B)\/(F/\E))\/((C/\B)\/(D/\E)))

Примеры формул не в ДНФ

(!(A/\B))

(A\/(B/\(C\/D)))

((!(A/\B))\/(C/\D))

((C\/B)/\(A\/D))

Описание программы и алгоритма

Программа включает в себя класс DNFQualifier, включающий в себя следующие методы: is_dnf(), initial_check(), check_parenthesis(), only_atomic_negations(), replace_special_syms(), apply_ranks_to_operations(), check_operations_order(), is_variable(), check_formula_syntax(), find_index_of_deepest_operation().

1. Метод is_dnf() проверяет, является ли данная формула ДНФ. Метод проверяет строку на содержание в ней только определённых символов, затем заменяет некоторые для последующей обработки и проводит остальные проверки. Метод сразу определит формулу как ДНФ, если формула атомарная, но не константа. Метод сразу определит формулу как не ДНФ, если она равна какой-либо из констант. Иначе, проводятся дополнительные проверки.
2. Метод initial_check() проводит первоначальную проверку формулы – проверка правильности расстановки скобок, соответствие количества скобок количеству логических операторов в формуле, проверяет формулу на соответствие синтаксису формул языка логики высказываний.
3. Метод check_parenthesis() - проводит проверку последовательности из скобок на правильность.
4. Метод only_atomic_negations() - проверяет формулу на наличие в ней отрицаний только атомарных формул.
5. Метод replace_special_syms() замещает некоторые из символов в формуле для упрощения машинной обработки.
6. Метод apply_ranks_to_operations() – соотносит каждую логическую операцию в формуле с уровнем подформулы, в которой она находится.
7. Метод check_operations_order() – проверяет последовательность логических операций в формуле на соответствие таковой в ДНФ.
8. Метод is_variable() – проверяет строку на соответствие её синтаксису переменной в языке логики высказываний.
9. Метод check_formula_syntax() – проверяет формулу на соответствие её синтаксису формулы языка логики высказываний.
10. Метод find_index_of_deepest_operation() – возвращает позицию символа, означающего логическую операцию, которая находится в подформуле самого высокого уровня.

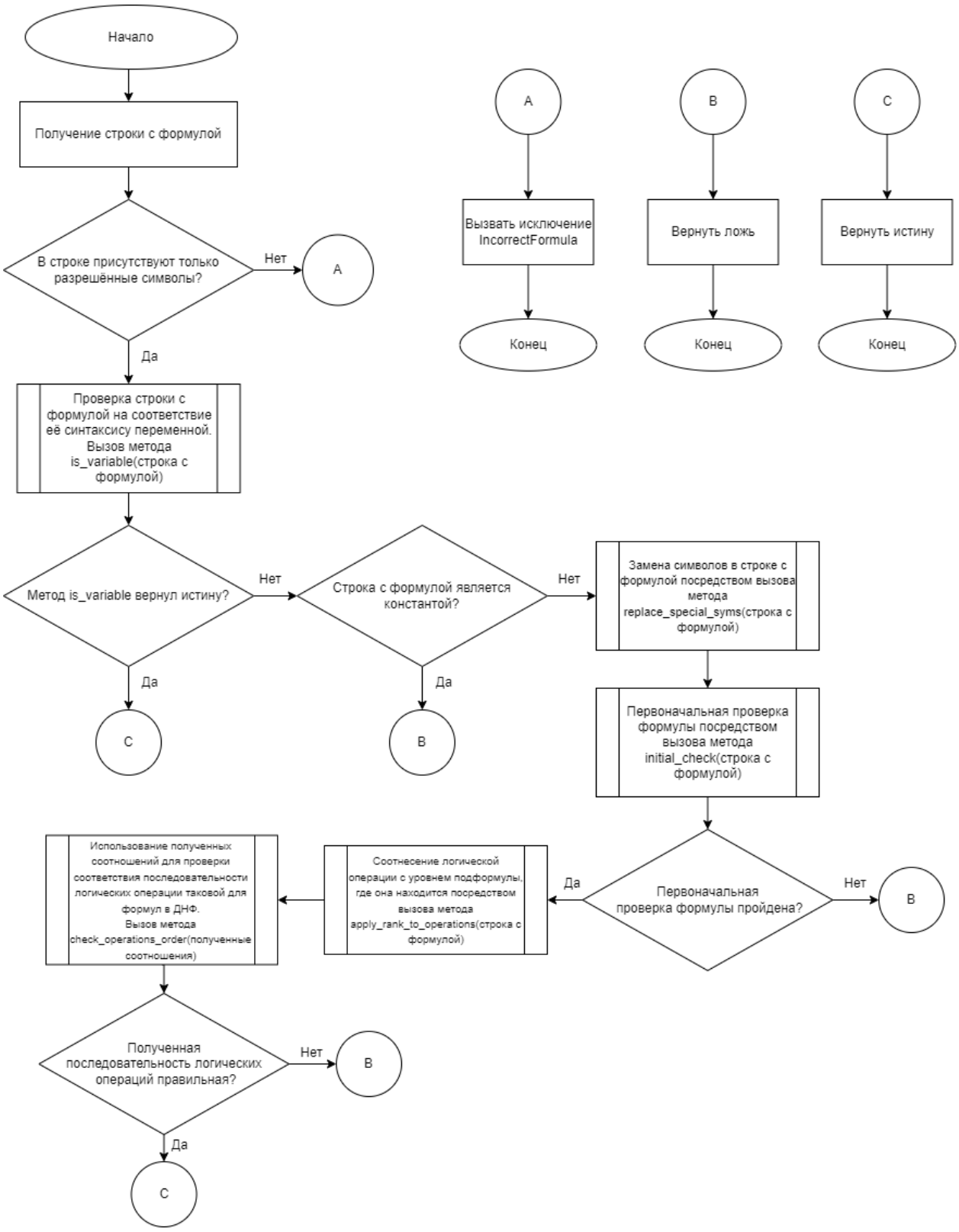

Рис. 1. Алгоритм метода is_dnf

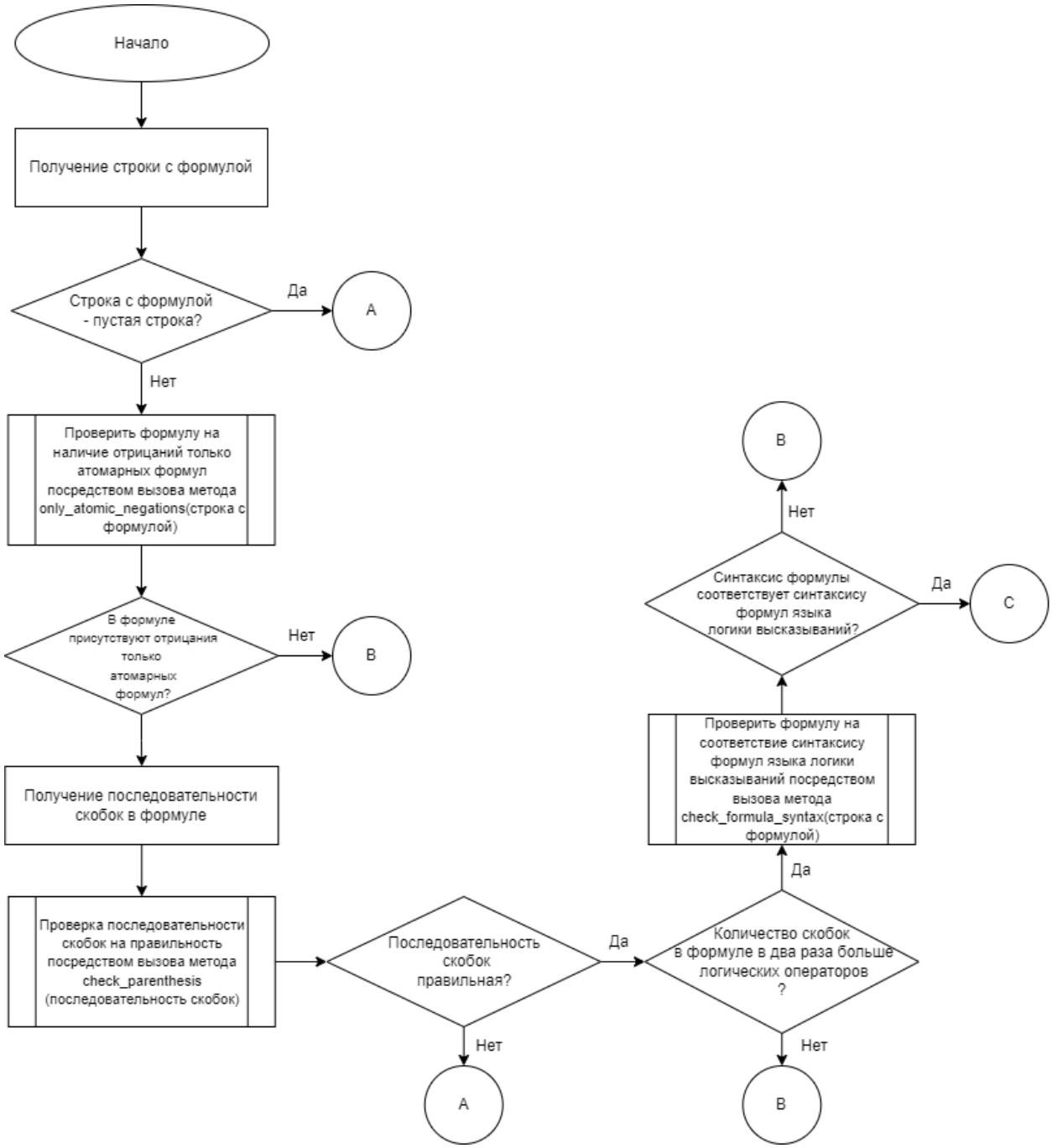

Рис 2. Алгоритм метода initial_check

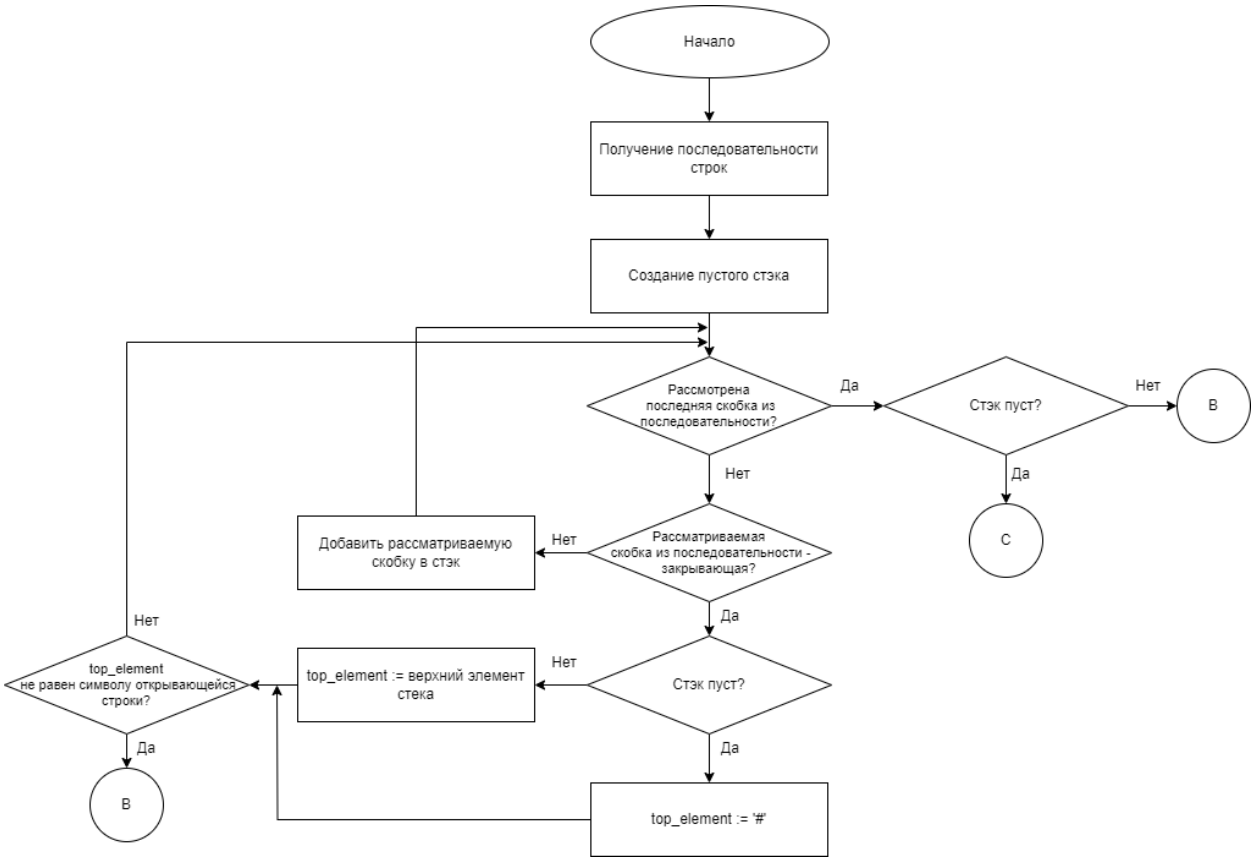

Рис. 3. Алгоритм метода check_parenthesis

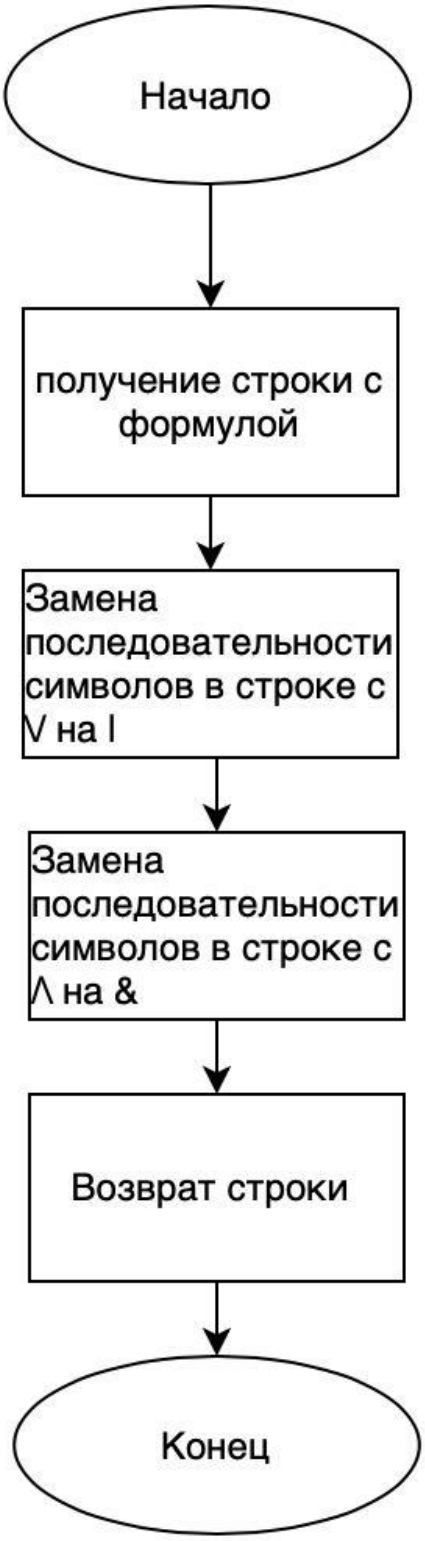

Рис. 4. Алгоритм метода replace_special_syms

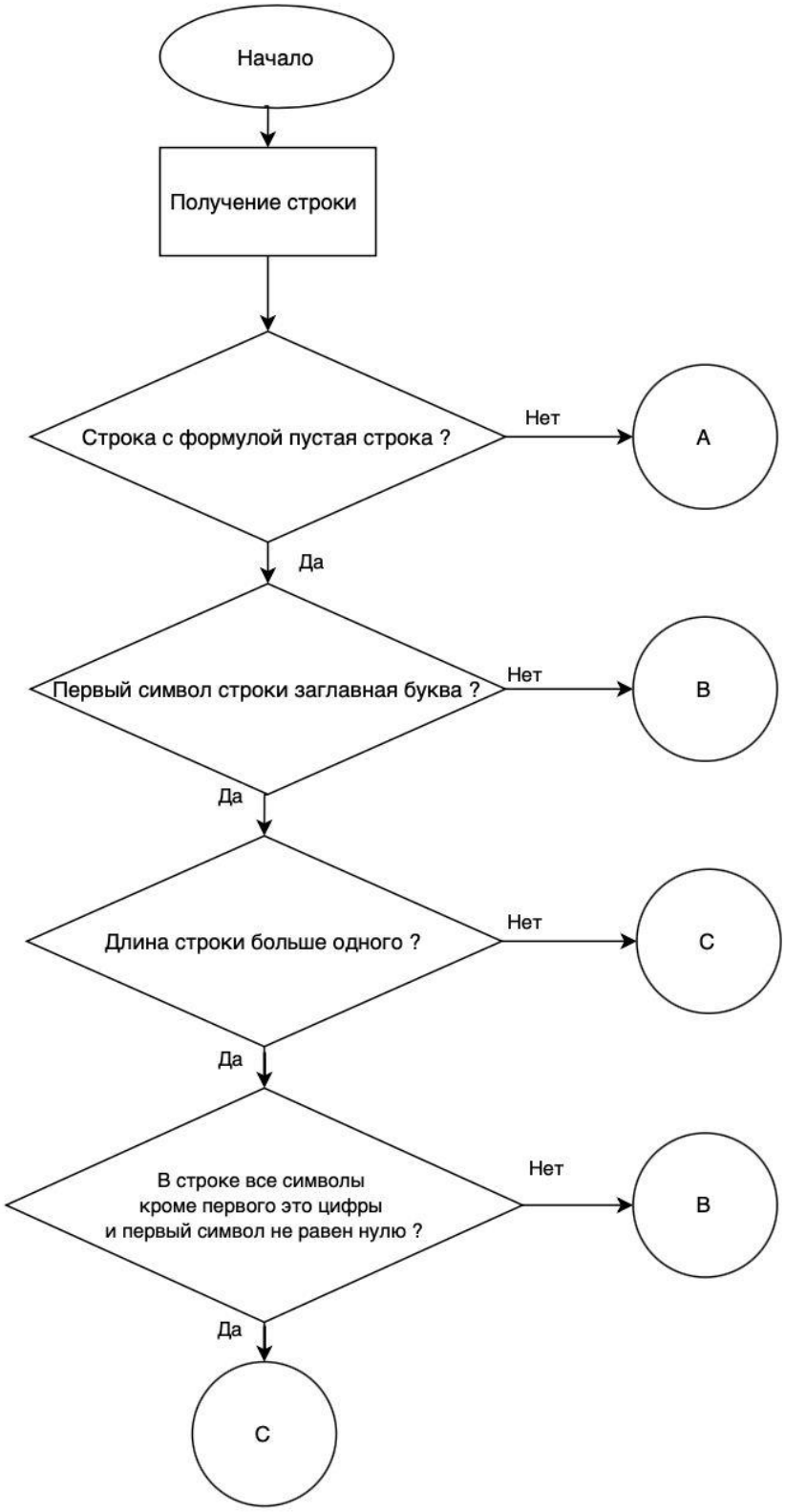

Рис. 5. Алгоритм метода is_variable

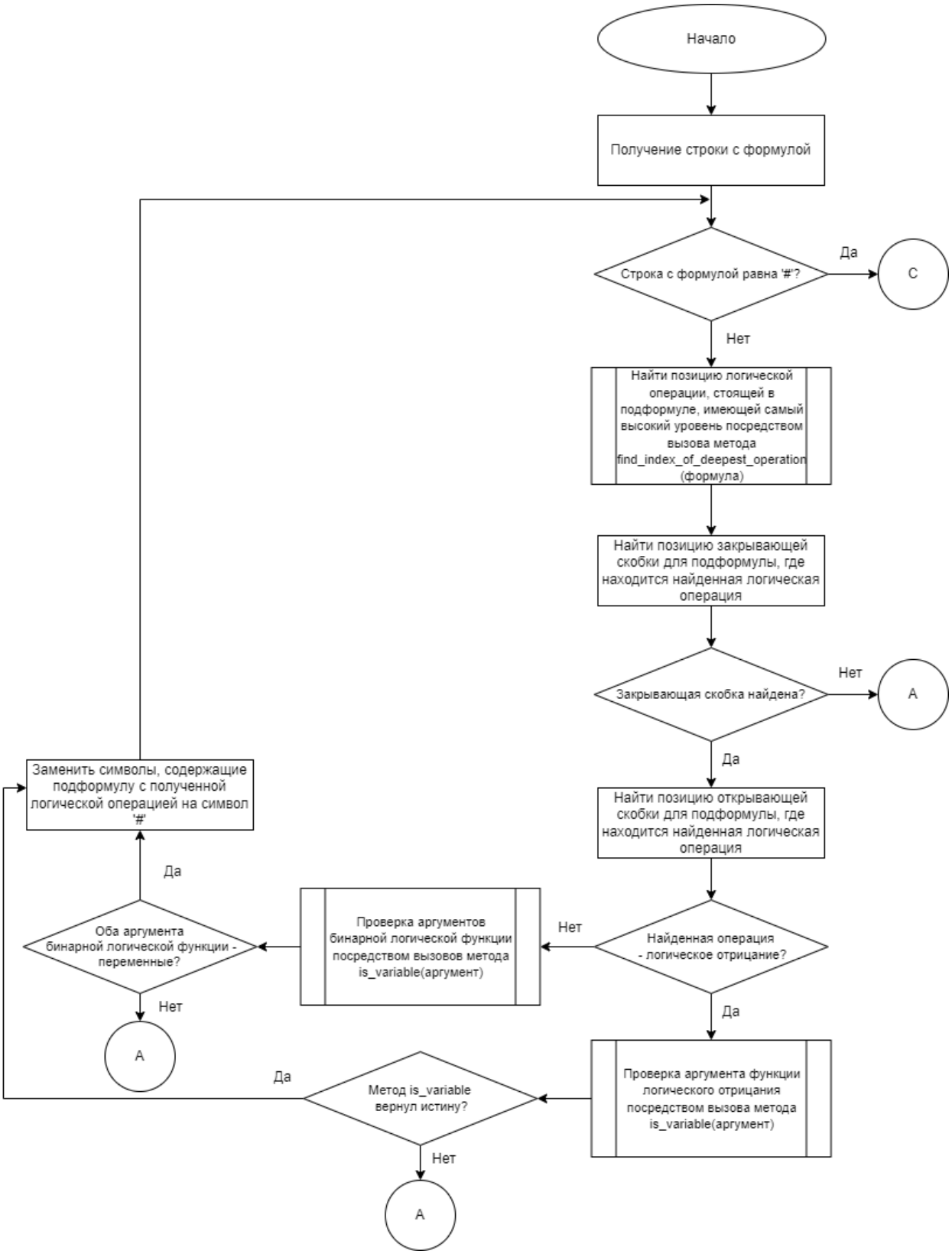

Рис. 6. Алгоритм метода check_formula_syntax

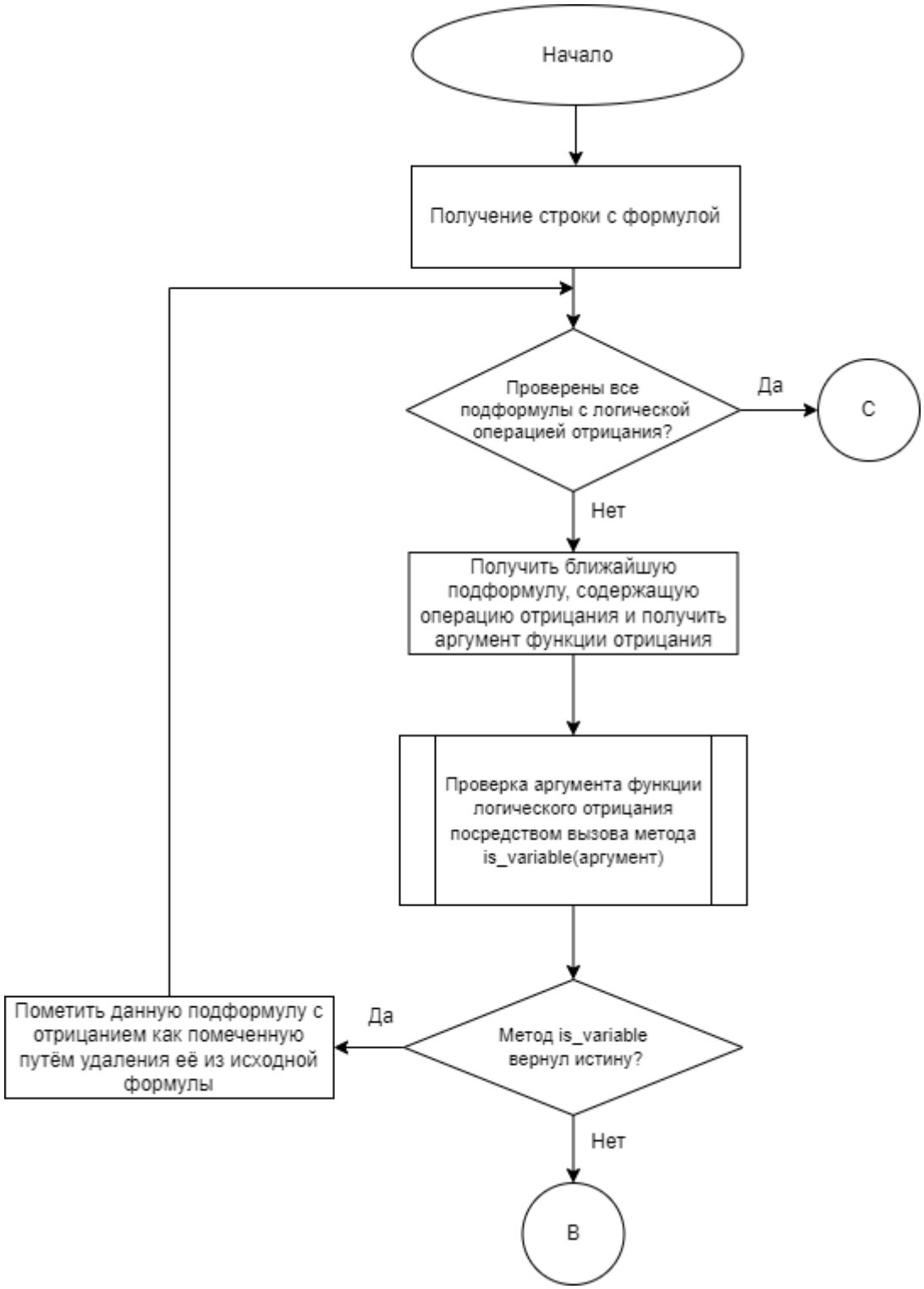

Рис. 7. Алгоритм метода only_atomic_negations

Рис. 8. Алгоритм метода find_index_of_deepest_operation

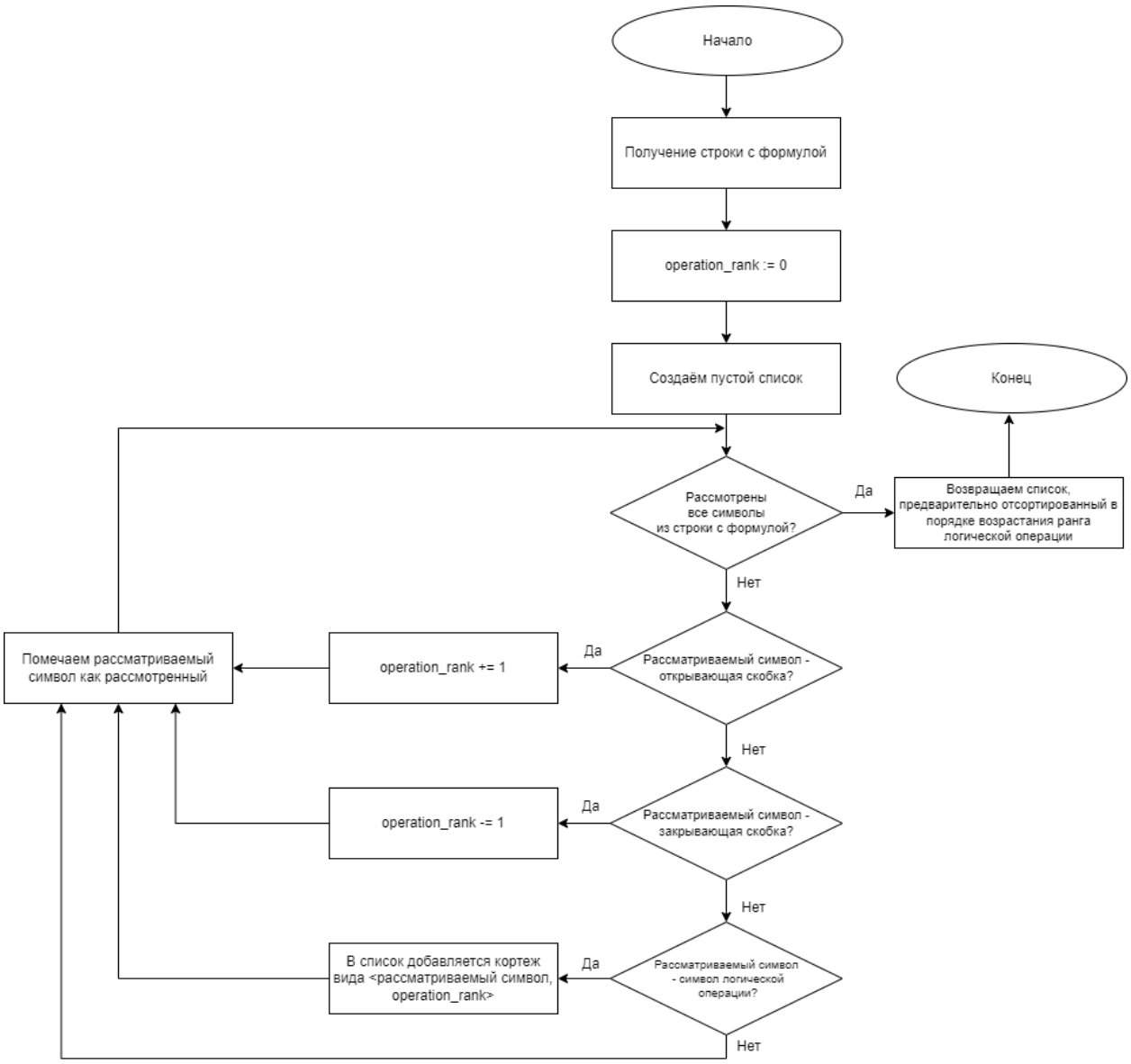

Рис. 9. Алгоритм метода apply_ranks_to_operations

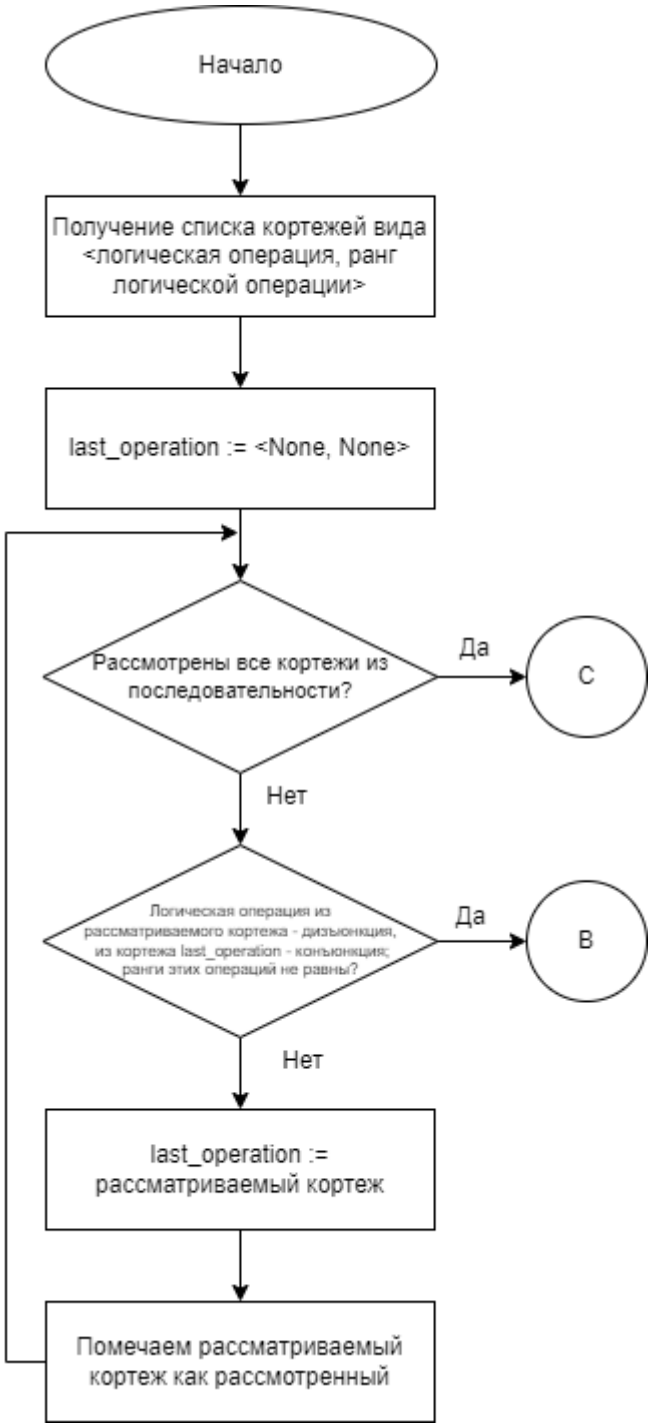

Рис. 10. Алгоритм метода check_operations_order

Тесты программы

```
Enter formula: (A\/B)
(A\/B) : True
```

Рис. 11. Тест 1

```
Enter formula: A
A : True
```

Рис. 12. Тест 2

```
Enter formula: ((C/\B)\/(D/\E))
((C/\B)\/(D/\E)) : True
```

Рис. 13. Тест 3

```
Enter formula: ((C/\B)\/((D/\E)\/G))
((C/\B)\/((D/\E)\/G)) : True
```

Рис. 14. Тест 4

```
Enter formula: (((A/\B)\/(F/\(E/\D)))\/((C/\B)\/(D/\E)))
(((A/\B)\/(F/\(E/\D)))\/((C/\B)\/(D/\E))) : True
```

Рис. 15. Тест 5

```
Enter formula: ((A/\B)
Error: Incorrect placement of parenthesis
```

Рис. 16. Тест 6

```
Enter formula: A/\B
A/\B : False
```

Рис 17. Тест 7

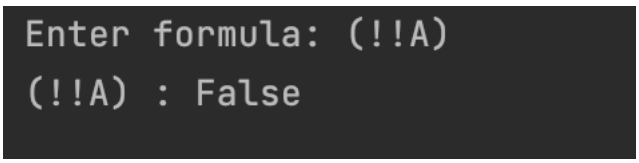

Рис 18. Тест 8

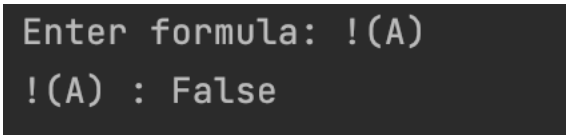

Рис 19. Тест 9

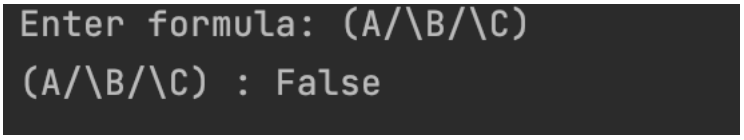

Рис 20. Тест 10

Вывод

В процессе выполнения лабораторной работы были приобретены навыки синтаксического анализа формул языка логики высказываний. Это было достигнуто через создание и реализацию алгоритмов, описанных ранее. Были разработаны блок-схемы для каждого из использованных алгоритмов, проведена отладка программы и осуществлено ручное тестирование окончательного результата.

Список использованных источников:

1. Логические основы интеллектуальных систем. Практикум : учеб.-метод. пособие / В. В. Голенков [и др.]. – Минск : БГУИР, 2011. – 70 с. : ил. ISBN 978-985-488-487-5.