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 7

Выполнил:

Студент гр. 321701 Самович В.М.

Проверил: Ивашенко В.П.

Минск 2025

**Тема:**

Представление и синтаксическая проверка формул языка логики высказываний.

**Цель:**

Приобрести навыки программирования алгоритмов синтаксического разбора формул языка логики высказываний.

**Задание:**

Перечислить фиктивные пропозициональные переменные в формуле сокращённого языка логики высказываний.

**Дополнительно:**

Предусмотреть работу системы в режиме тестирования знаний пользователя.

**Грамматика языка логики высказываний:**

<константа>::=0|1

<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>

**Схемы функций программы:**

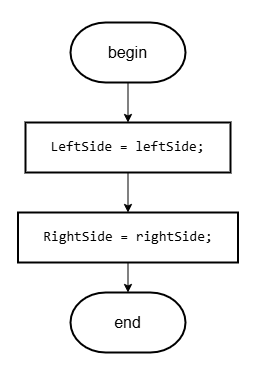


Рис 1. Конструктор Conjunction(IEvaluatable leftSide, IEvaluatable rightSide) класса Conjunction.

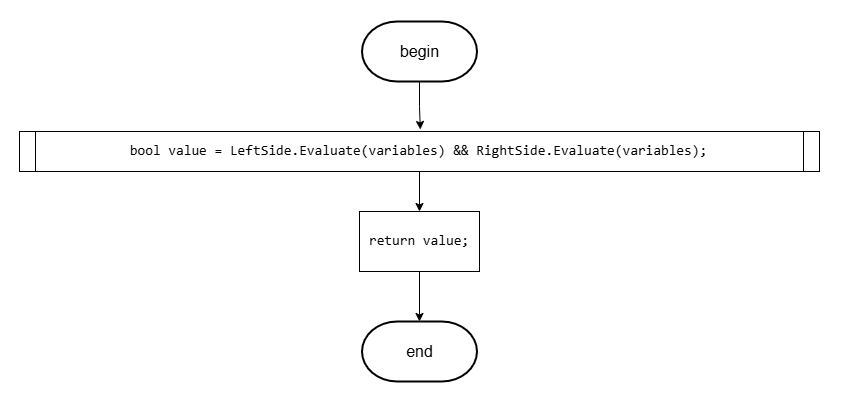


Рис 2. Метод Evaluate(EvaluatableSource variables) класса Conjunction.

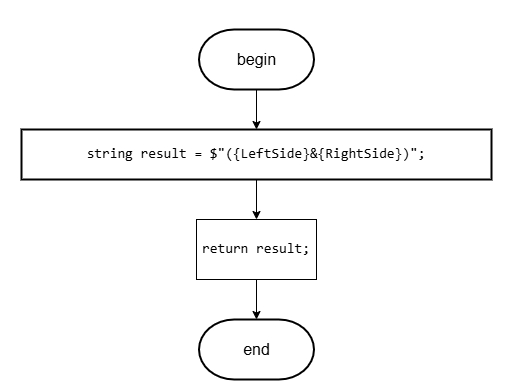


Рис 3. Метод ToString() класса Conjunction.

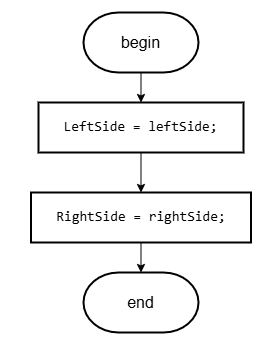


Рис 4. Конструктор Disjunction(IEvaluatable leftSide, IEvaluatable rightSide) класса Disjunction.

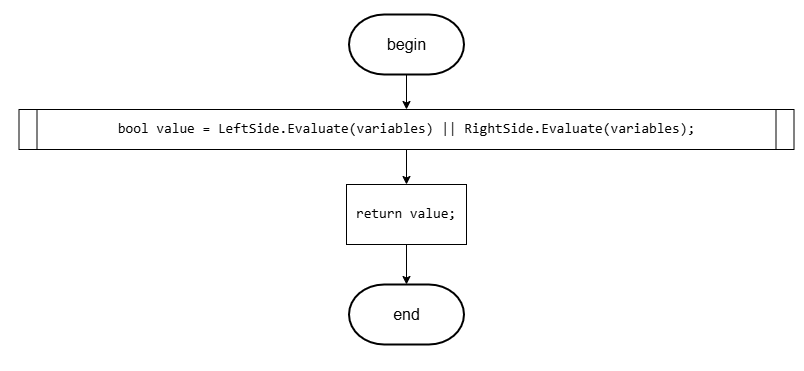


Рис 5. Метод Evaluate(EvaluatableSource variables) класса Disjunction.

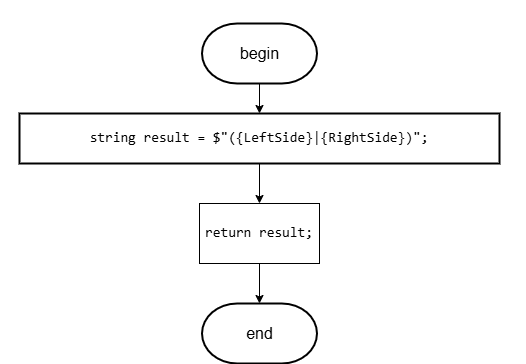


Рис 6. Метод ToString() класса Disjunction.

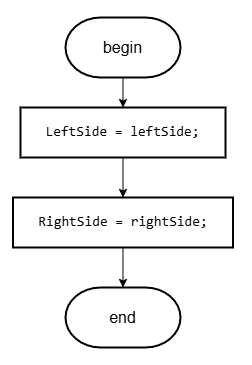


Рис 7. Конструктор Equivalence(IEvaluatable leftSide, IEvaluatable rightSide) класса Equivalence.

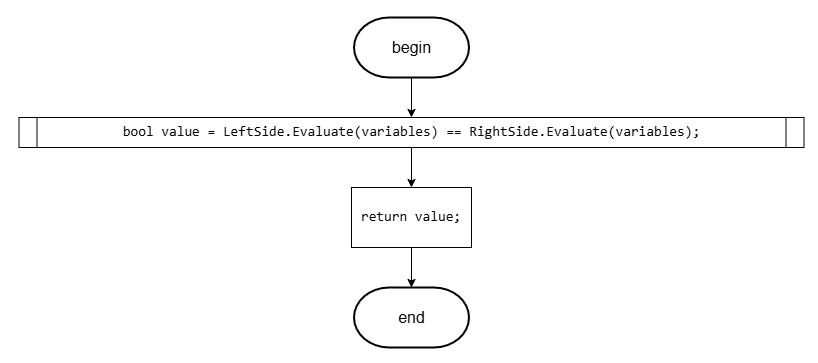


Рис 8. Метод Evaluate(EvaluatableSource variables) класса Equivalence.

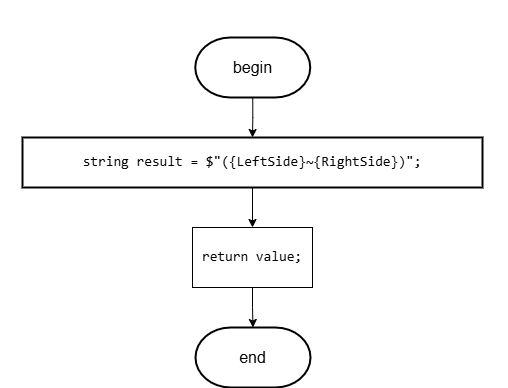


Рис 9. Метод ToString() класса Equivalence.

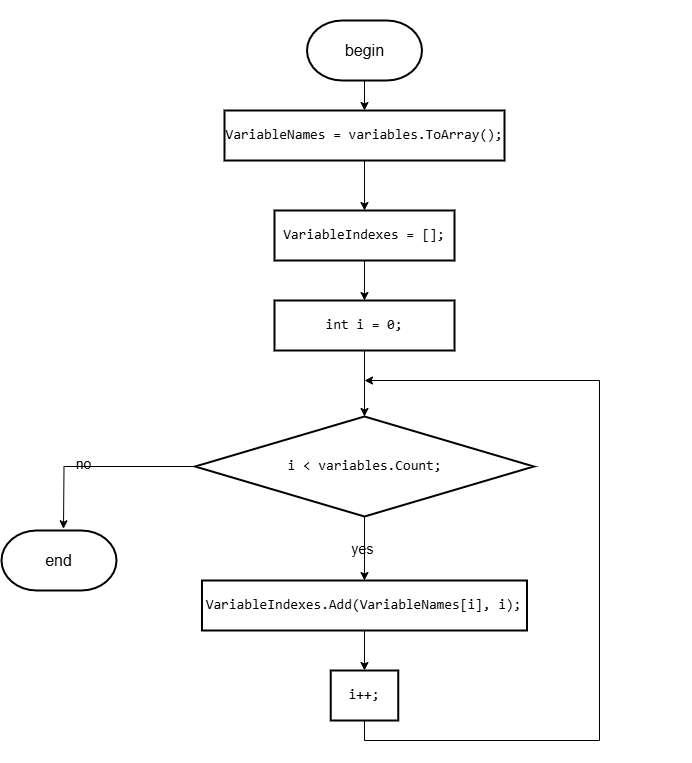


Рис 10. Конструктор EvaluatableSource(List<string> variables) класса EvaluatableSource.

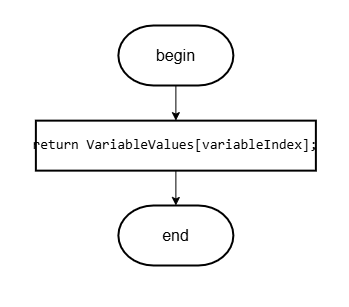


Рис 11. Метод this[int variableIndex] класса EvaluatableSource.

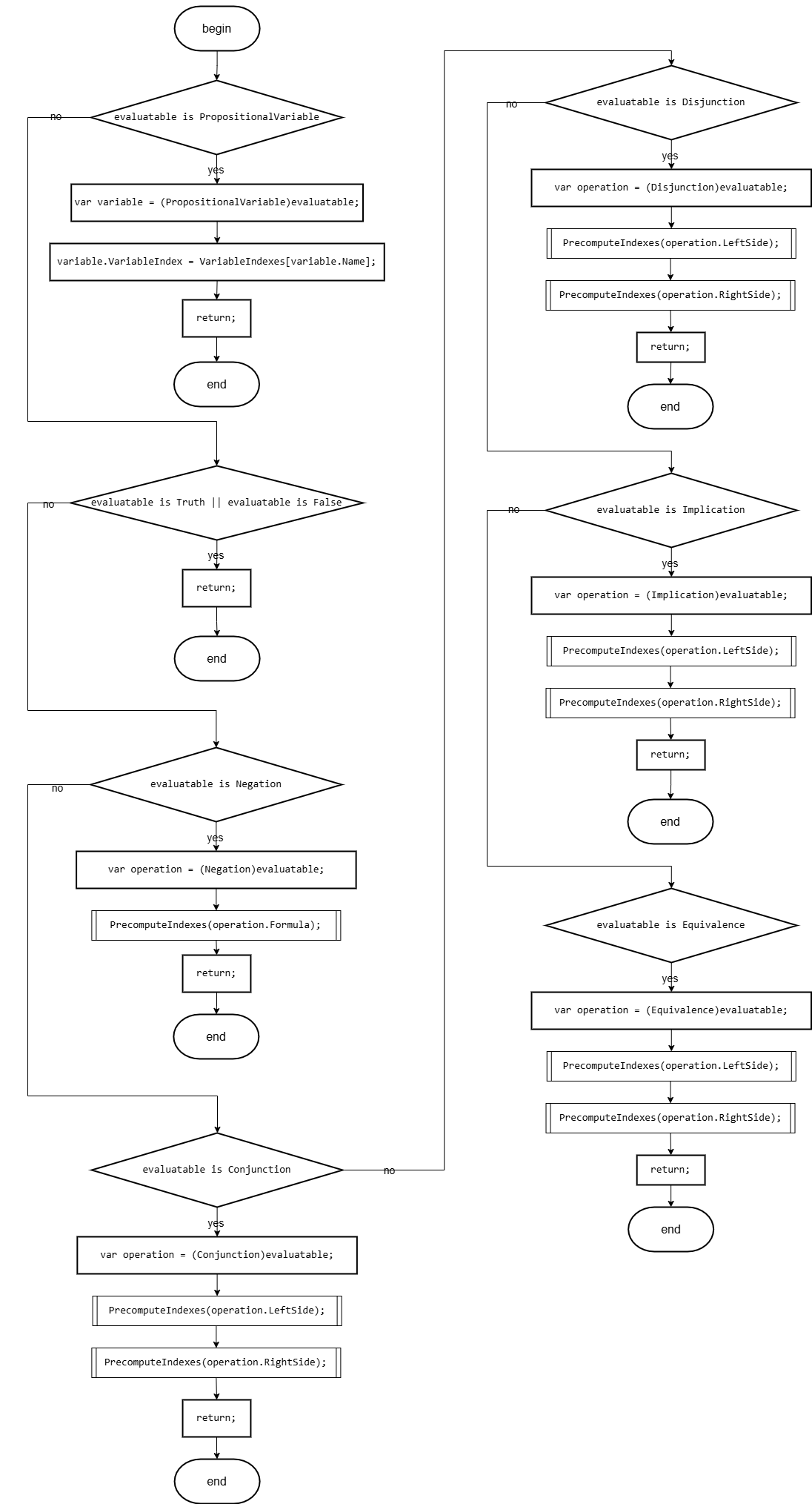


Рис 12. Метод PrecomputeIndexes(IEvaluatable evaluatable) класса EvaluatableSource.

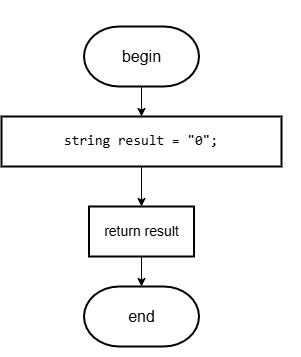


Рис 13. Метод ToString() класса False.

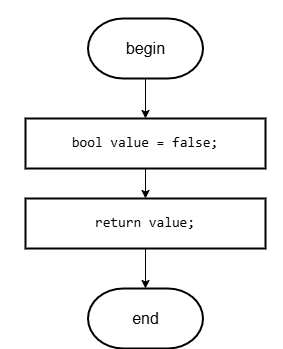


Рис 14. Метод Evaluate(EvaluatableSource variables) класса False.

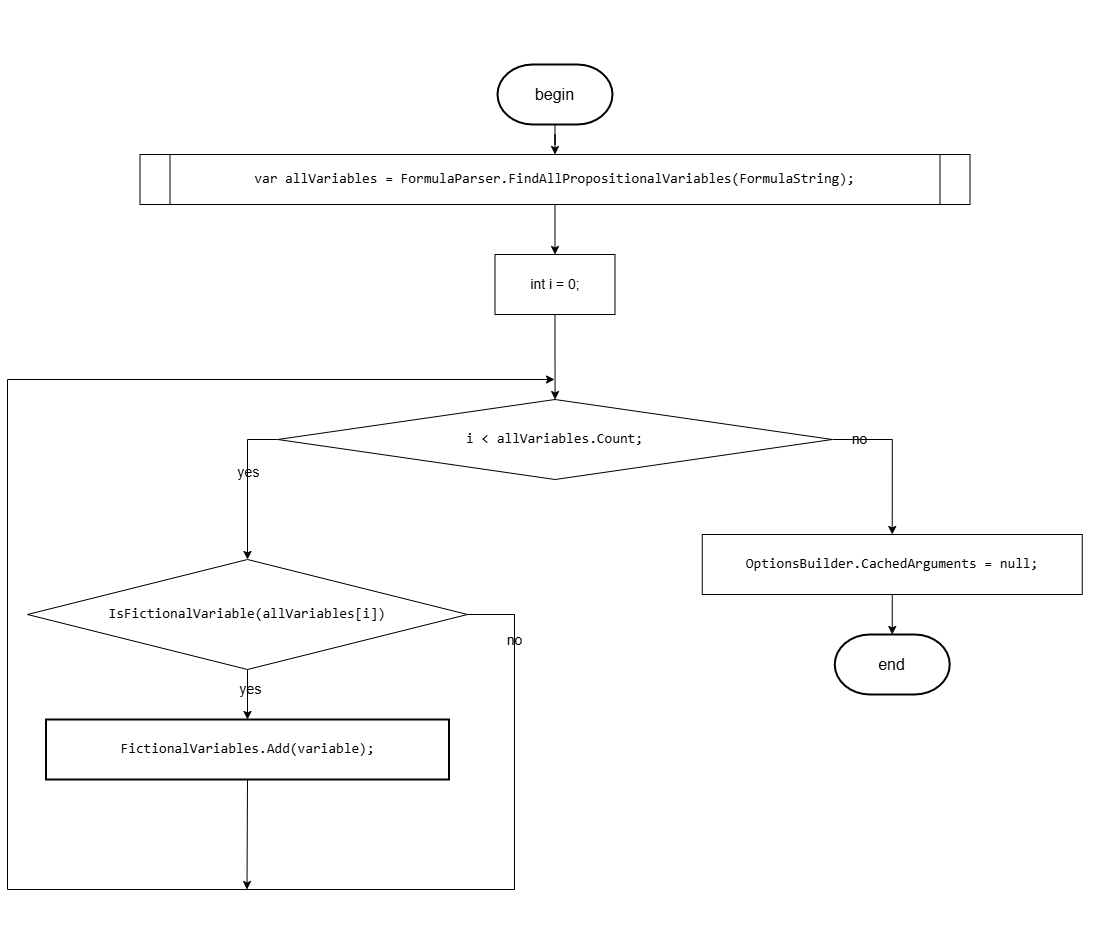


Рис 15. Метод FindFictionalVariables(bool showTime = true) класса FictionalVariablesFinder.

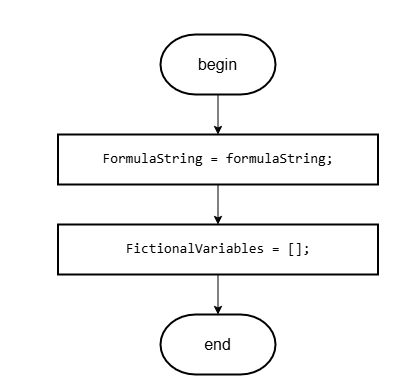


Рис 16. Конструктор FictionalVariablesFinder (string formulaString) класса FictionalVariablesFinder.

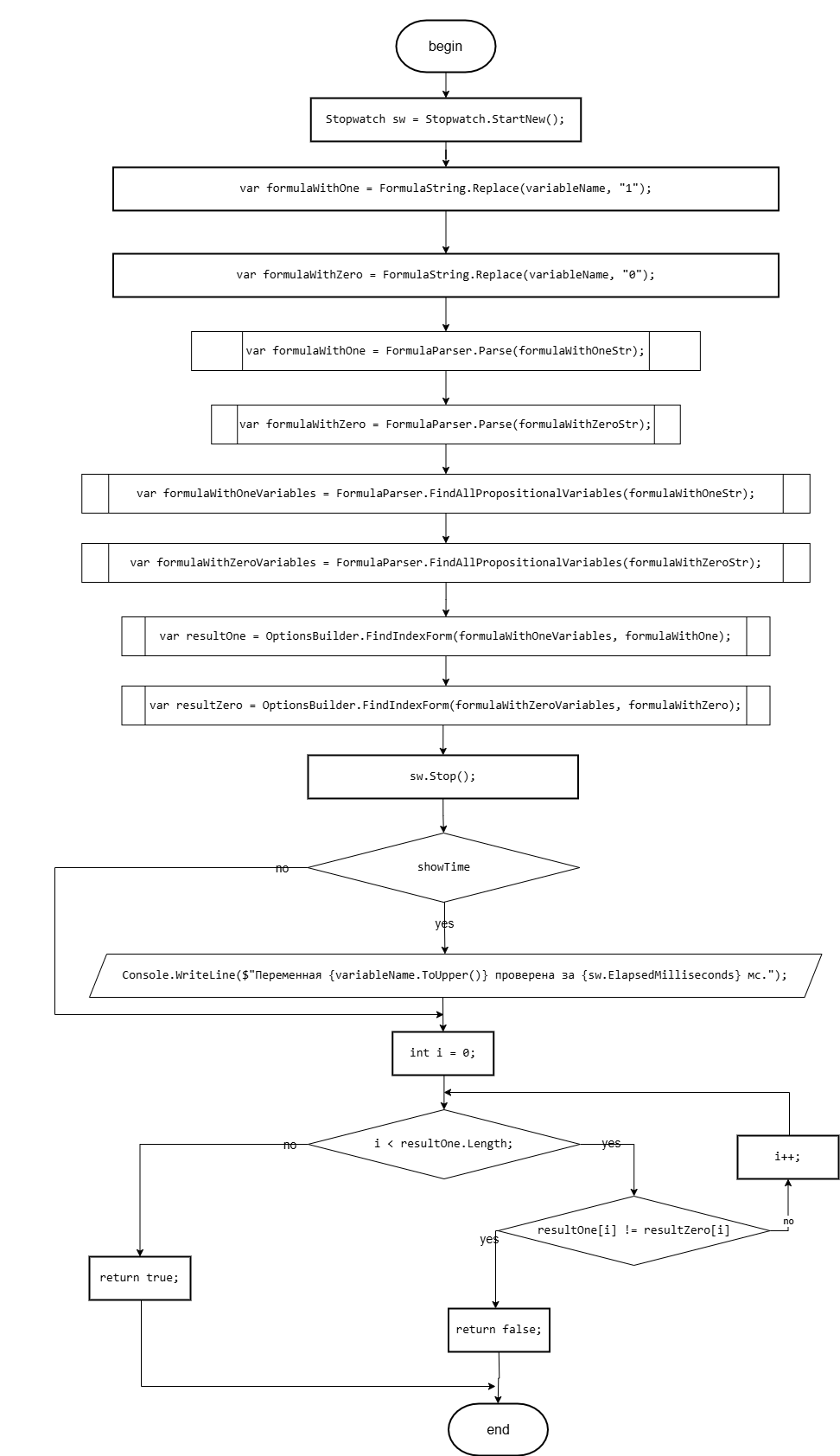


Рис 17. Метод IsFictionalVariable(string variableName, bool showTime = true) класса FictionalVariablesFinder.

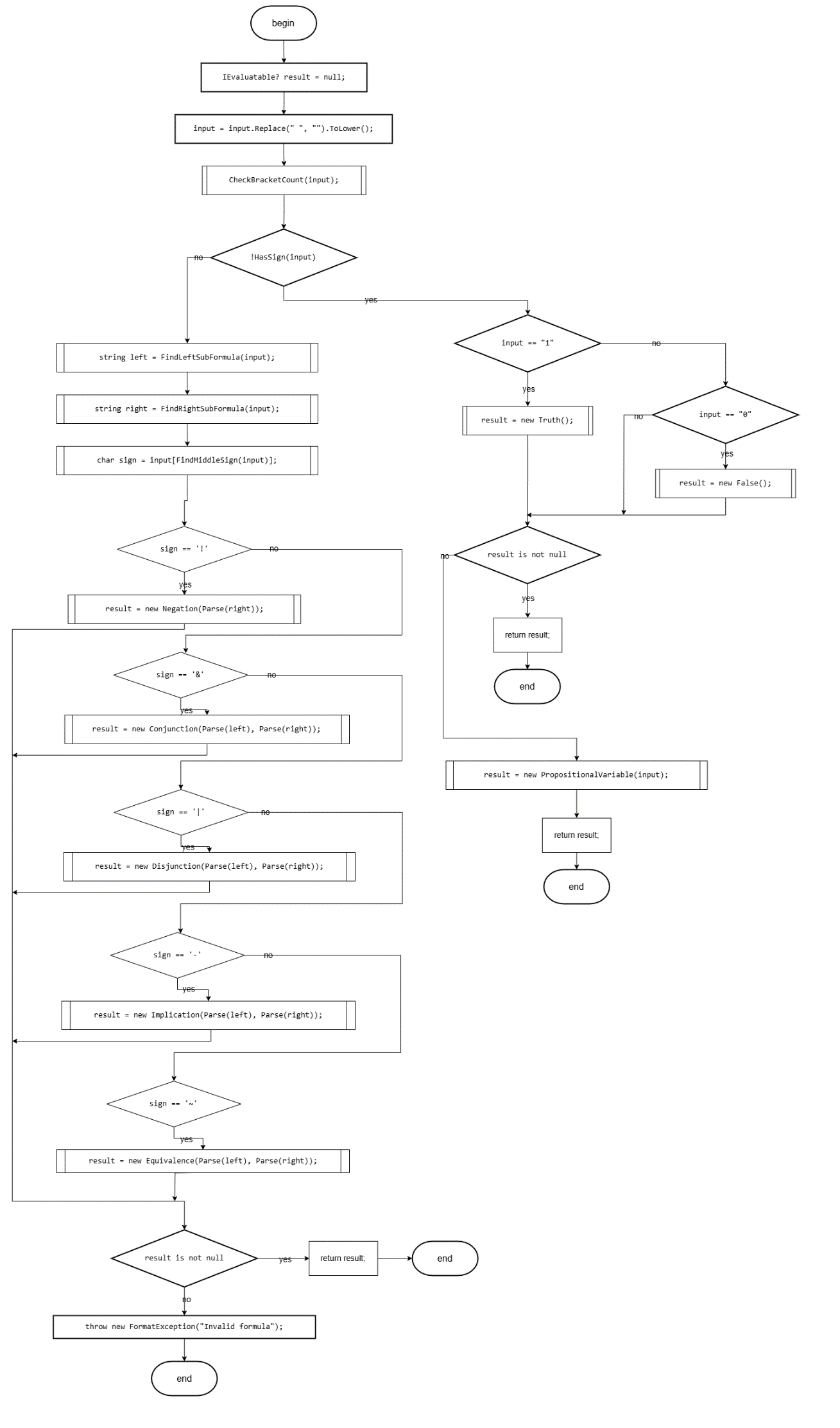


Рис 18. Метод Parse(string input) класса FormulaParser.

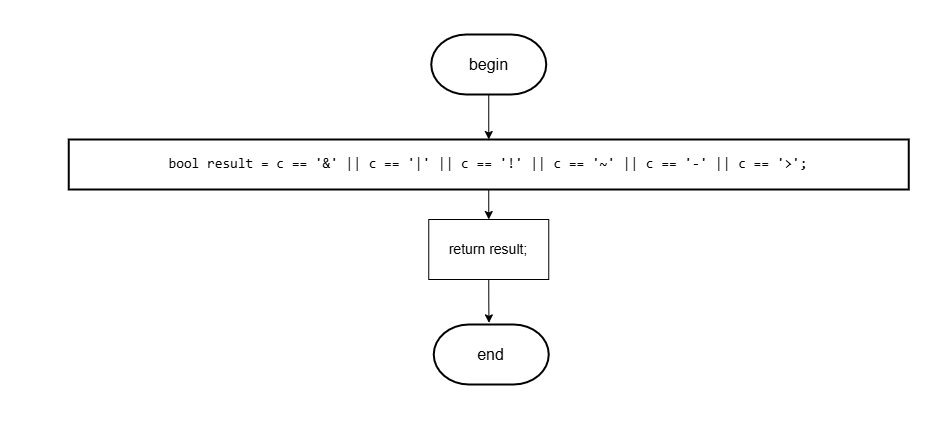


Рис 19. Метод IsLogicalSign(char c) класса FormulaParser.

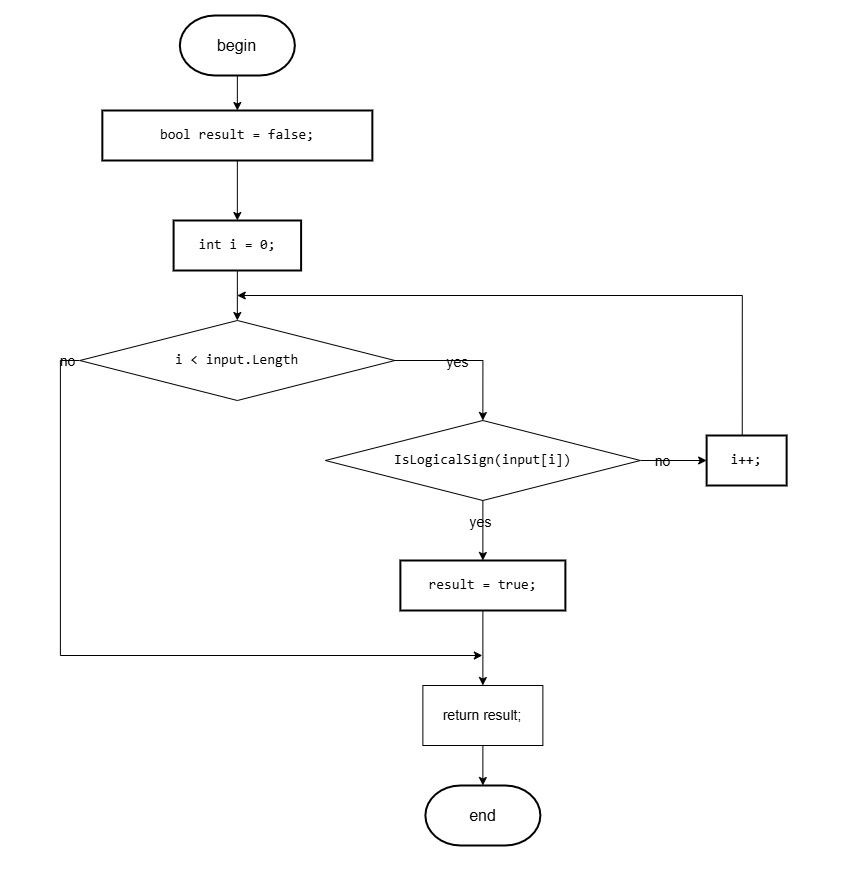


Рис 20. Метод HasSign(string input) класса FormulaParser.

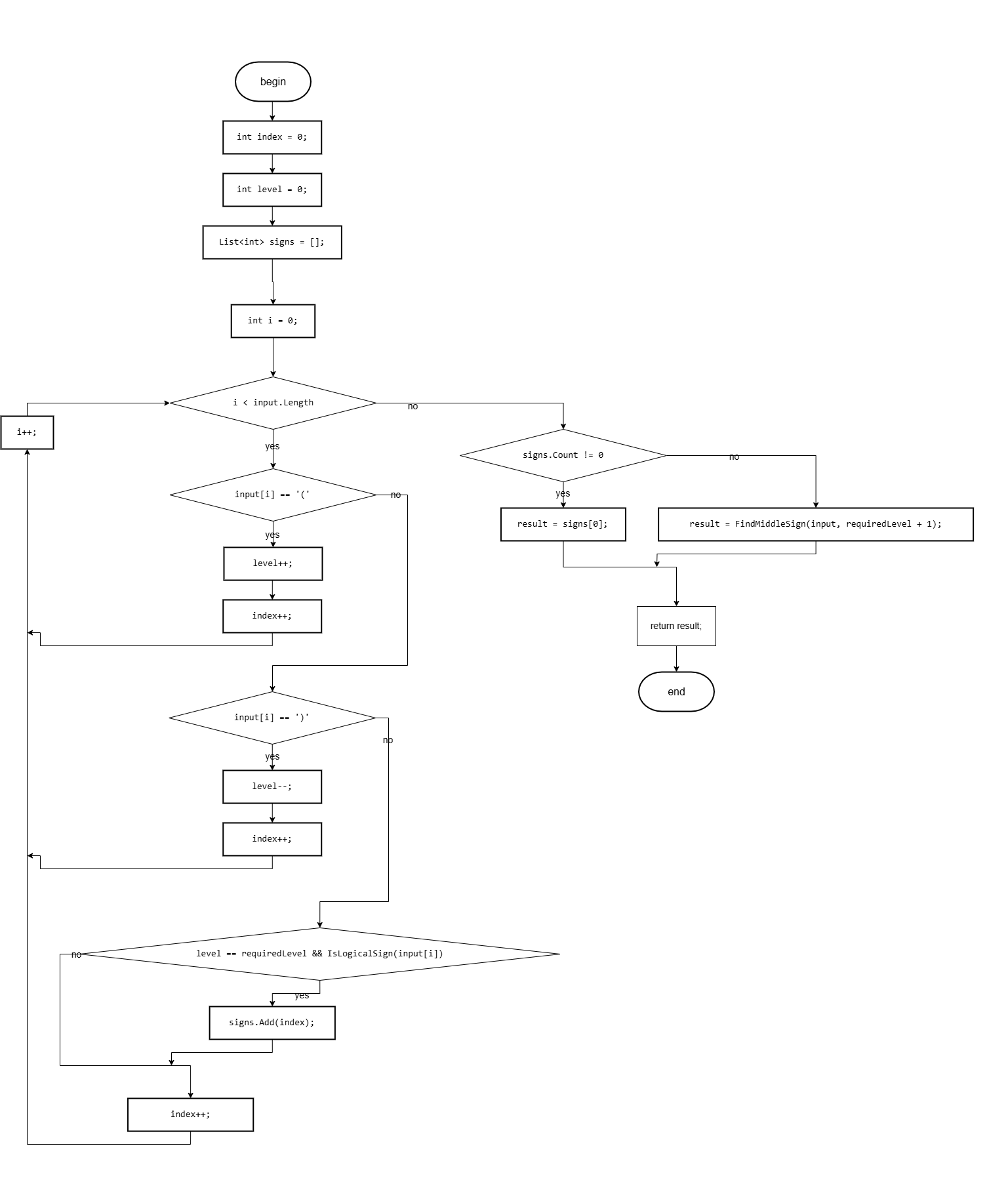


Рис 21. Метод FindMiddleSign(string input, int requiredLevel = 0) класса FormulaParser.

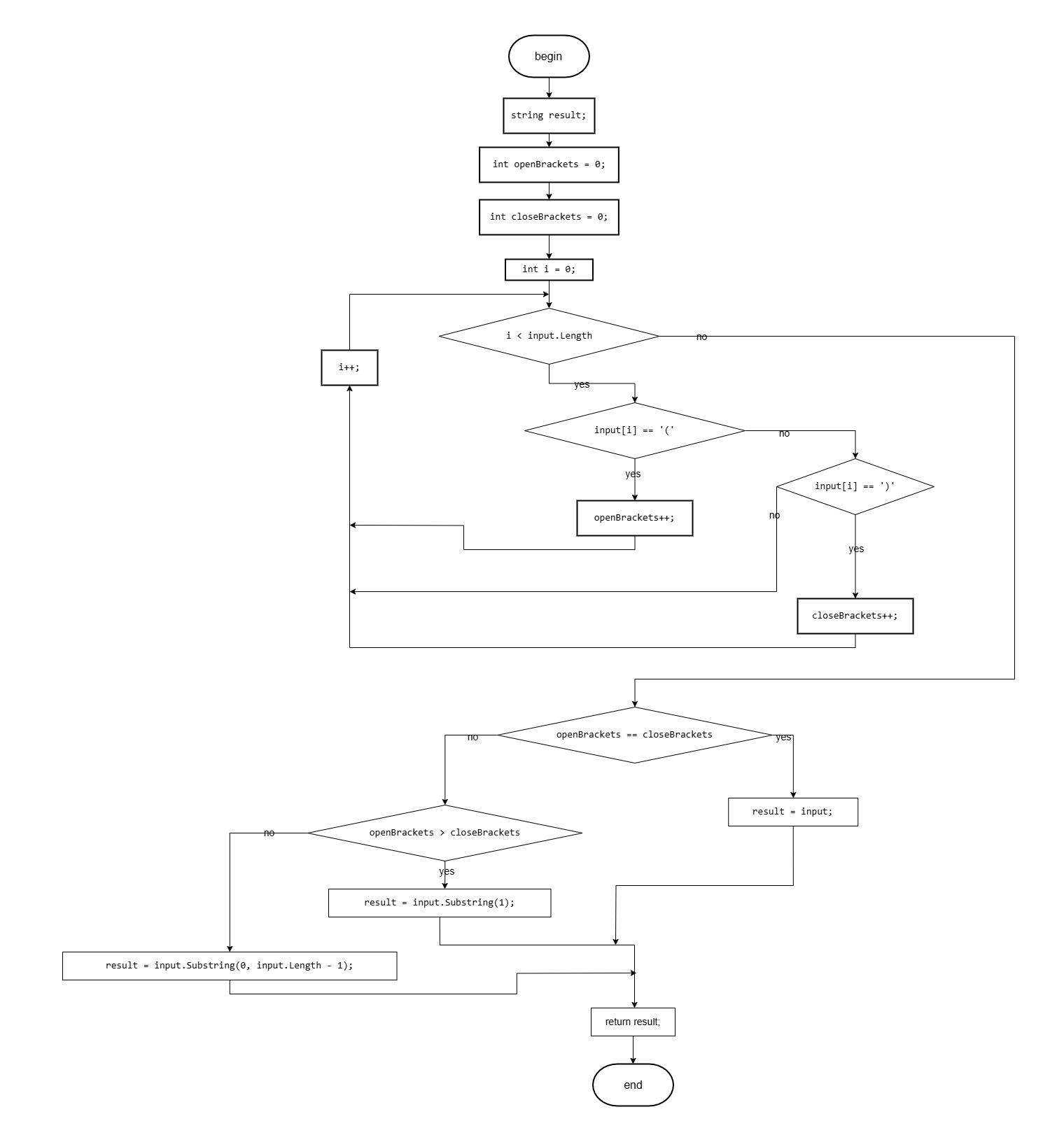


Рис 22. Метод NormalizeBrackets(string input) класса FormulaParser.

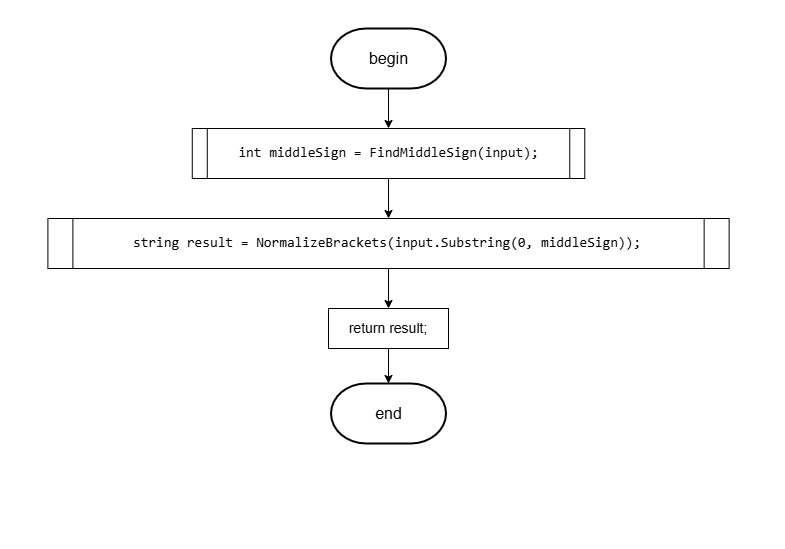


Рис 23. Метод FindLeftSubformula(string input) класса FormulaParser.

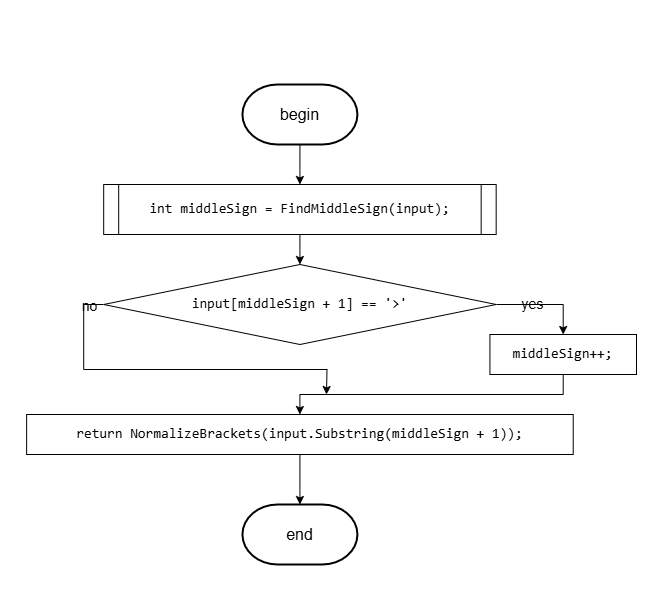


Рис 24. Метод FindRightSubformula(string input) класса FormulaParser.

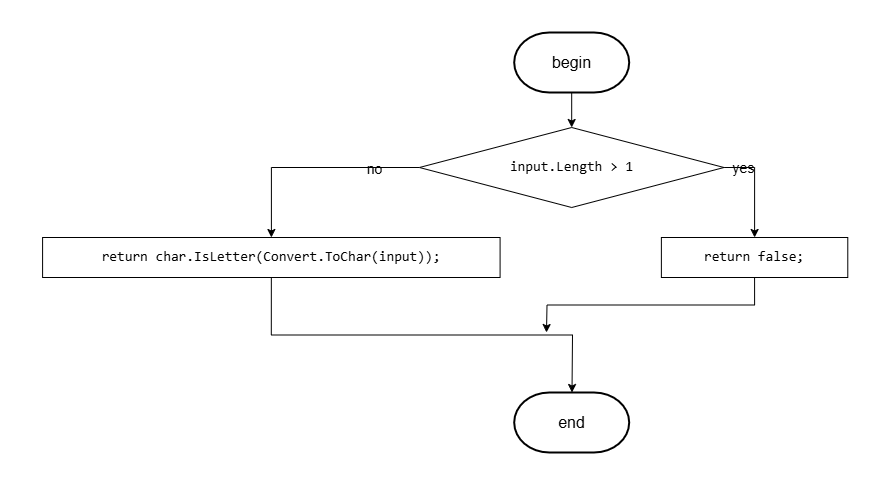


Рис 25. Метод IsPropositionalVariable(string input) класса FormulaParser.

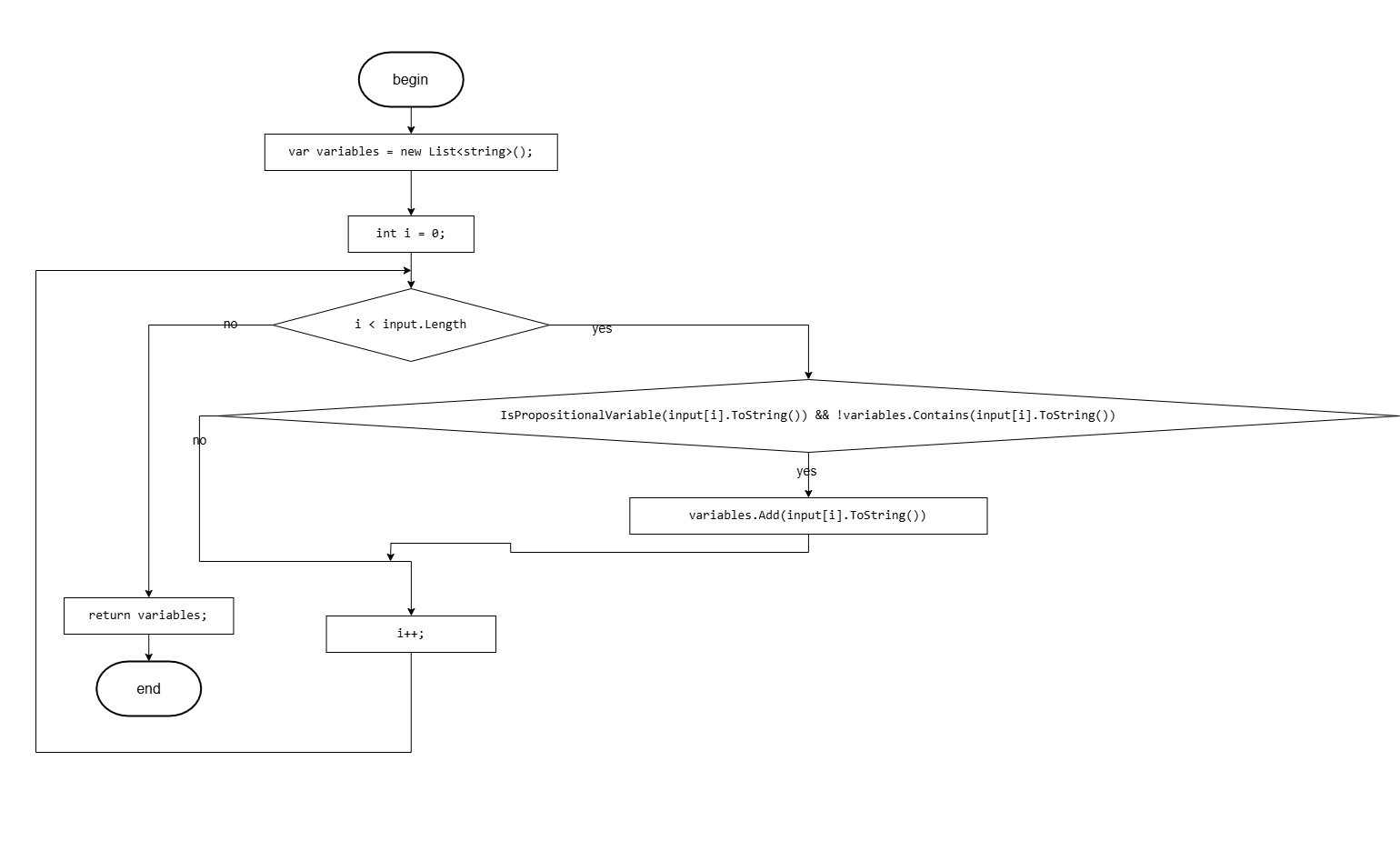


Рис 26. Метод FindAllPropositionalVariables(string input) класса FormulaParser.

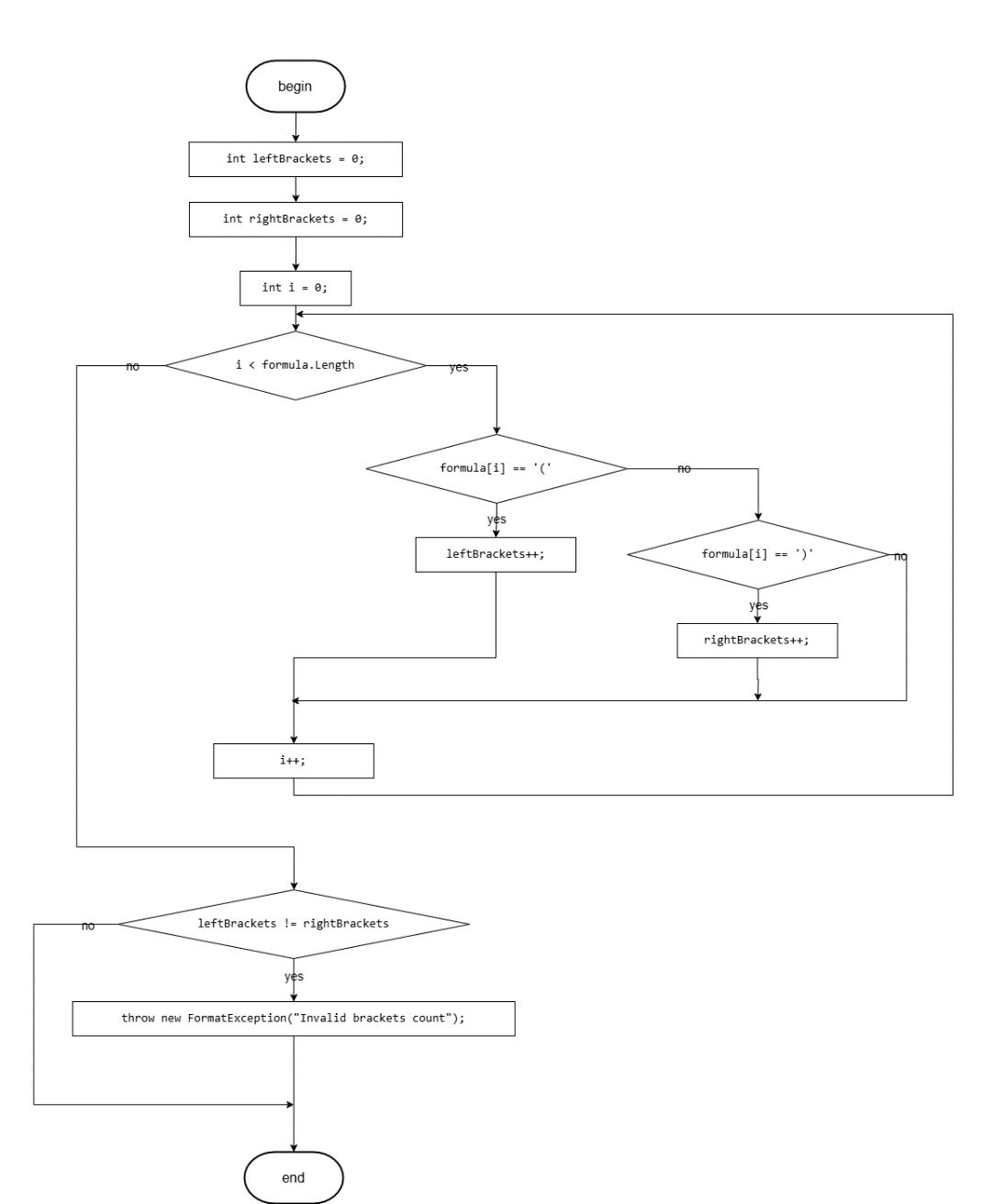


Рис 27. Метод CheckBracketCount(string formula) класса FormulaParser.

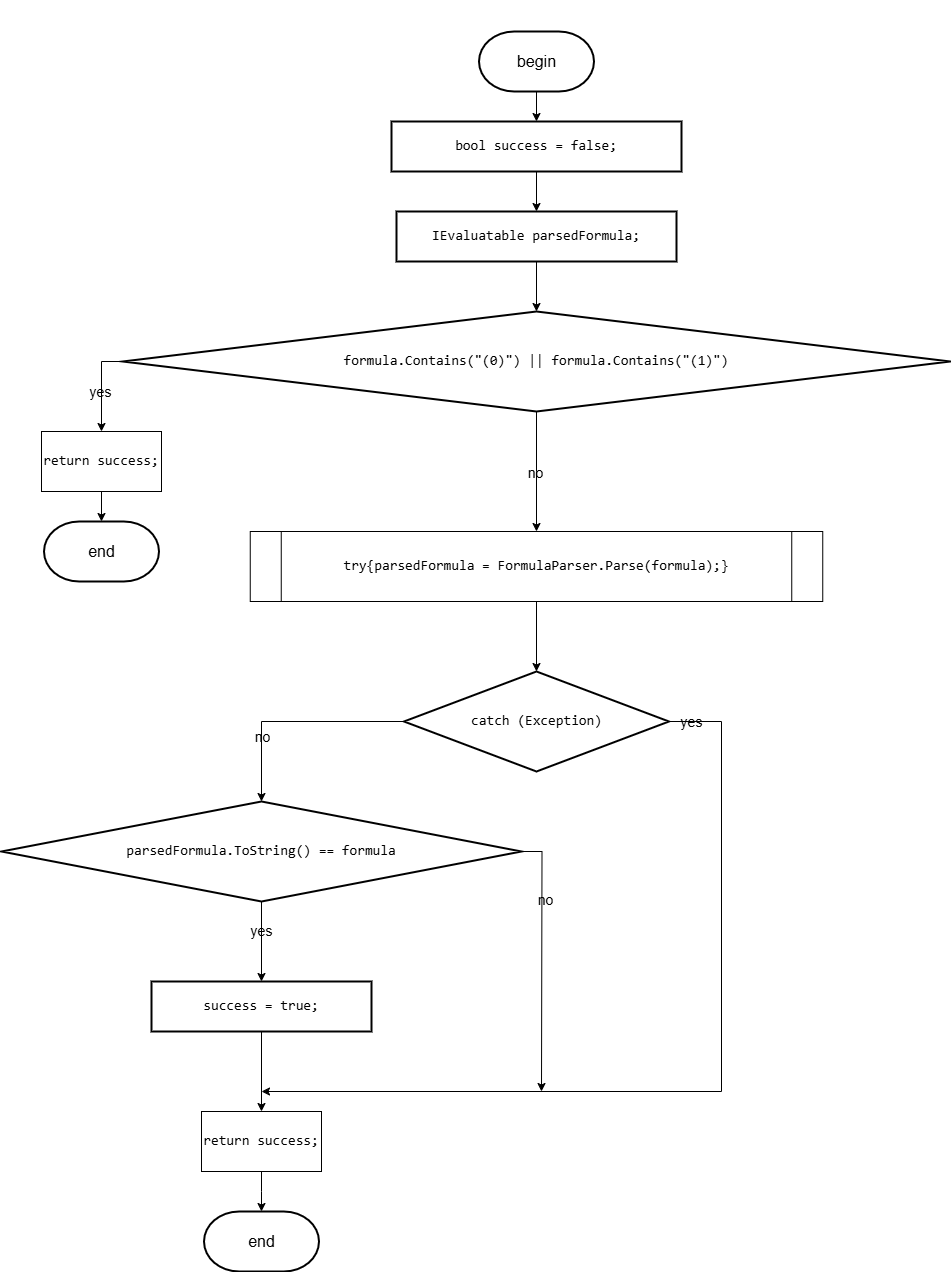


Рис 28. Метод Check(string formula) класса FormulaStringChecker.

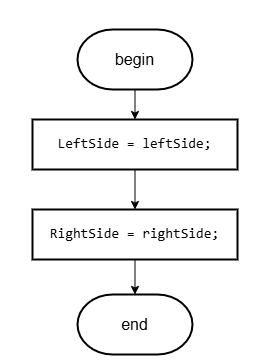


Рис 29. Конструктор Implication(IEvaluatable leftSide, IEvaluatable rightSide) класса Implication.

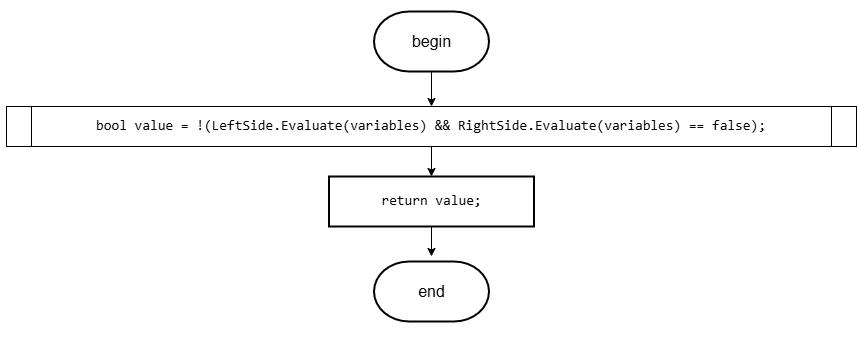


Рис 30. Метод Evaluate(EvaluatableSource variables) класса Implication.

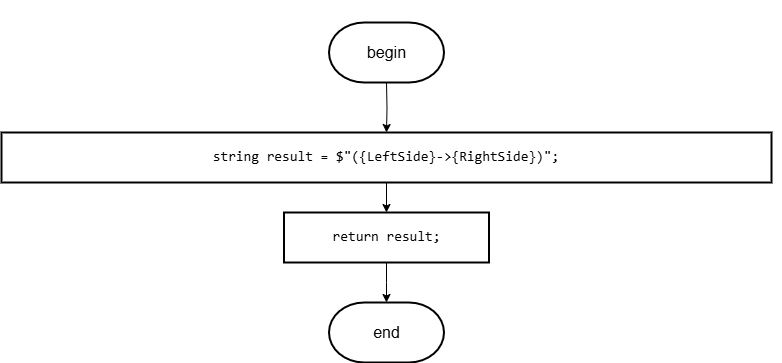


Рис 31. Метод ToString() класса Implication.

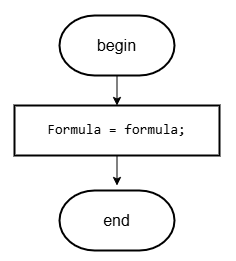


Рис 32. Конструктор Negation(IEvaluatable formula) класса Negation.

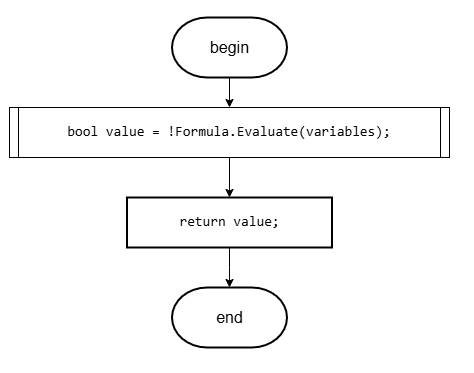


Рис 33. Метод Evaluate(EvaluatableSource variables) класса Negation.

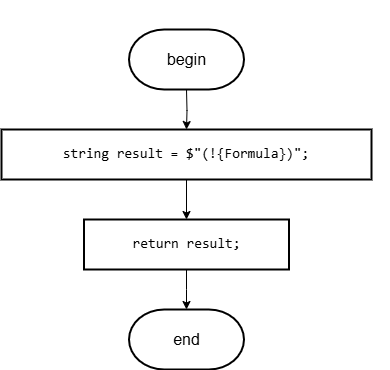


Рис 34. Метод ToString() класса Negation.

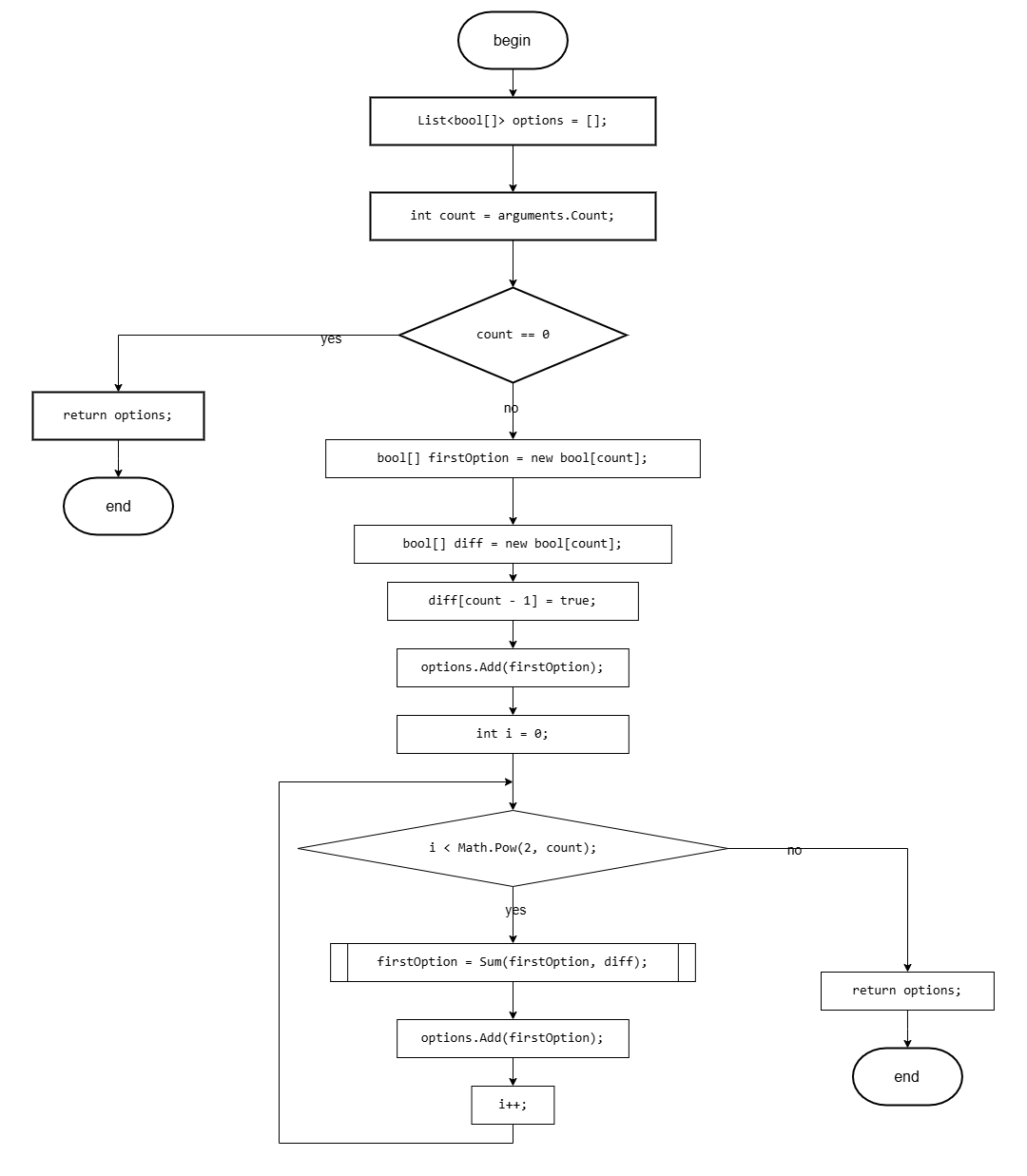


Рис 35. Метод BuildArguments(List<string> arguments) класса OptionsBuilder.

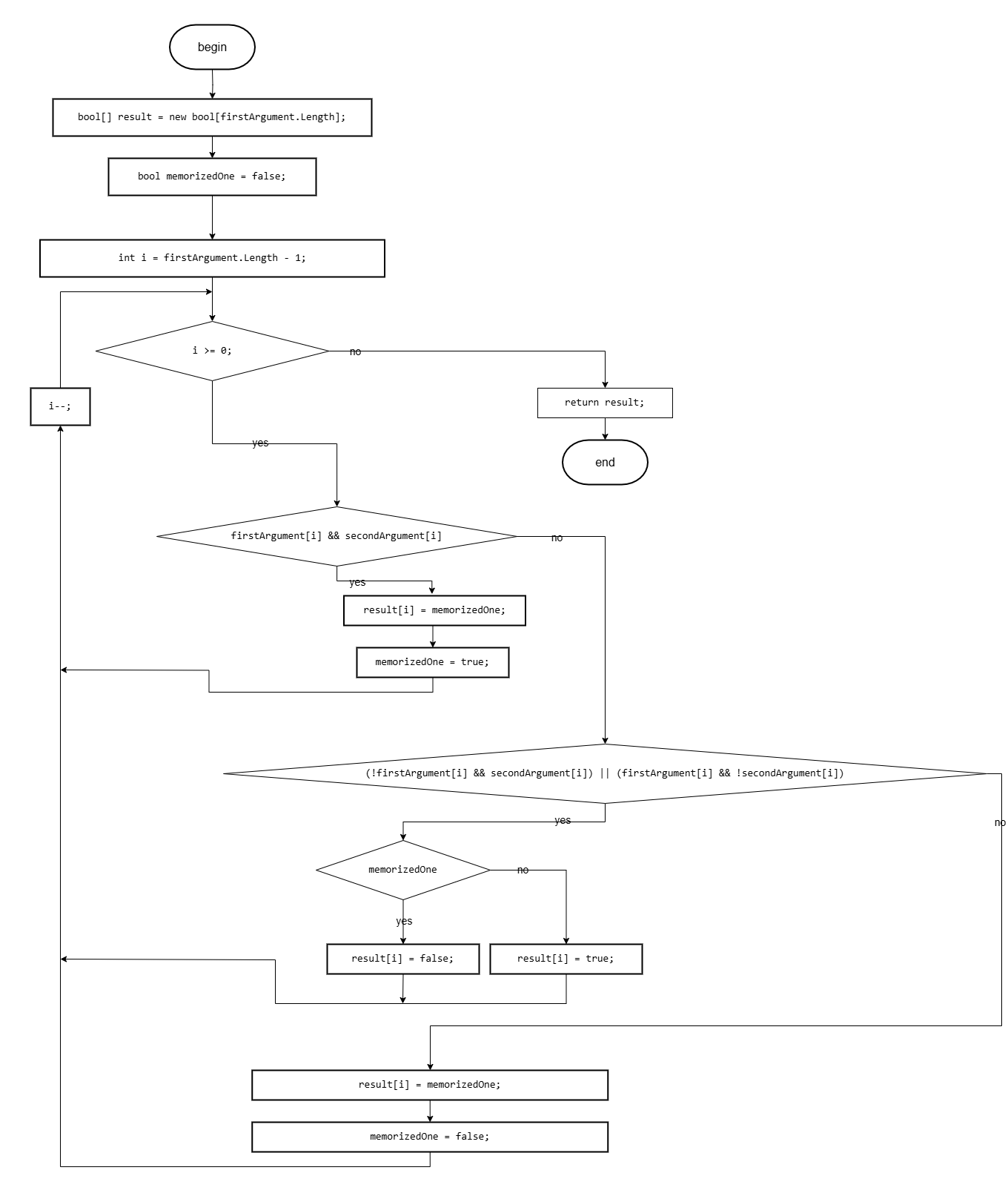


Рис 36. Метод Sum(bool[] firstArgument, bool[] secondArgument) класса OptionsBuilder.

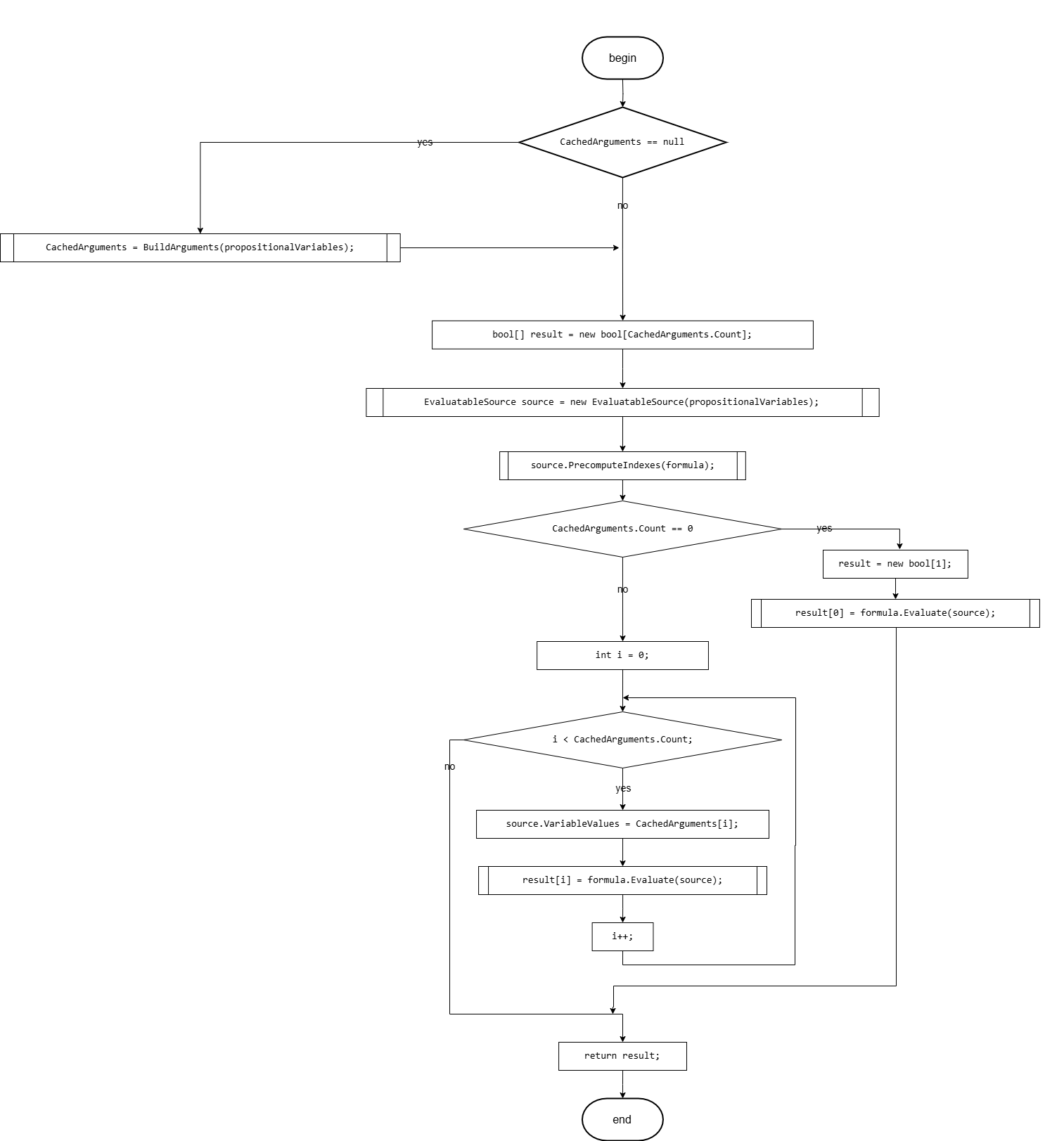


Рис 37. Метод FindIndexForm(List<string> propositionalVariables, IEvaluatable formula) класса OptionsBuilder.

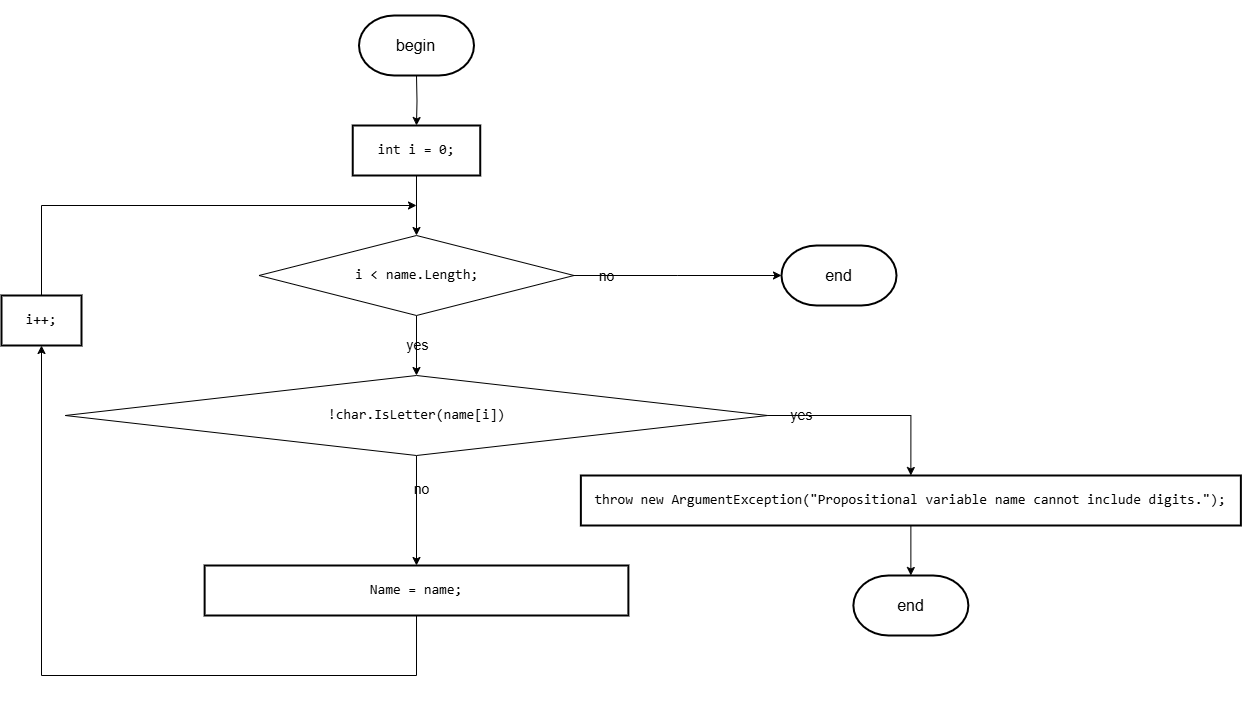


Рис 38. Конструктор PropositionalVariable(string name) класса PropositionalVariable.

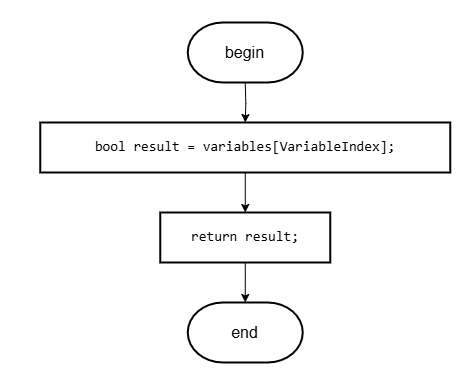


Рис 39. Метод Evaluate(EvaluatableSource variables) класса PropositionalVariable.

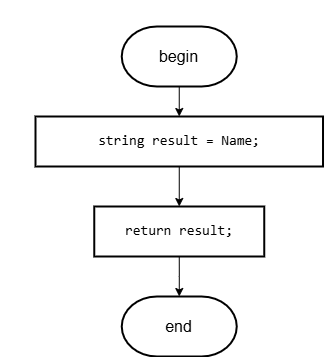


Рис 40. Метод ToString() класса PropositionalVariable.

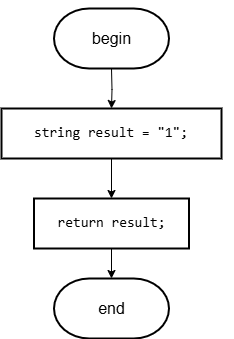


Рис 41. Метод ToString() класса Truth.

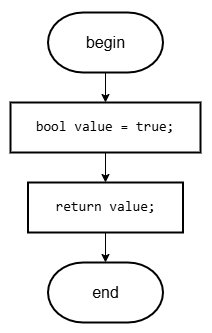


Рис 42. Метод Evaluate(EvaluatableSource variables) класса Truth.

**Программная реализация:**

Для выполнения данной лабораторной работы был использован язык программирования C# (.NET 9). В программе используются такие структуры данных, как список, словарь, дерево. Логическая формула представляется в программе в структуре данных дерево, элементами которого являются сложные формулы, атомарные формулы, переменные и константы.

После запуска программы пользователь выбирает режим работы: поиск фиктивных переменных или проверку знаний пользователя.

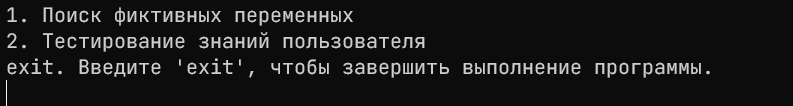


Рис. 43. Выбор режима работы программы

Если пользователь выбрал режим поиска фиктивных переменных, то далее программа попросит пользователя ввести логическую формулу. Если введённая строка не является формулой сокращённого языка логики высказываний, то программа выдаст пользователю сообщение об ошибке.

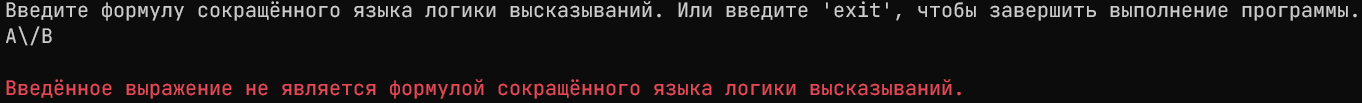


Рис. 44. Поведение программы, если пользователь ввёл некорректную строку

Далее пользователь сможет продолжить поиск фиктивных переменных до того, как не введёт «exit». Для каждой введённой формулы программа будет перечислять найденные фиктивные переменные, а если они отсутствуют, то выдавать сообщение «Фиктивных переменных не найдено». После сообщения, также выводится время, затраченное на поиск фиктивных переменных.

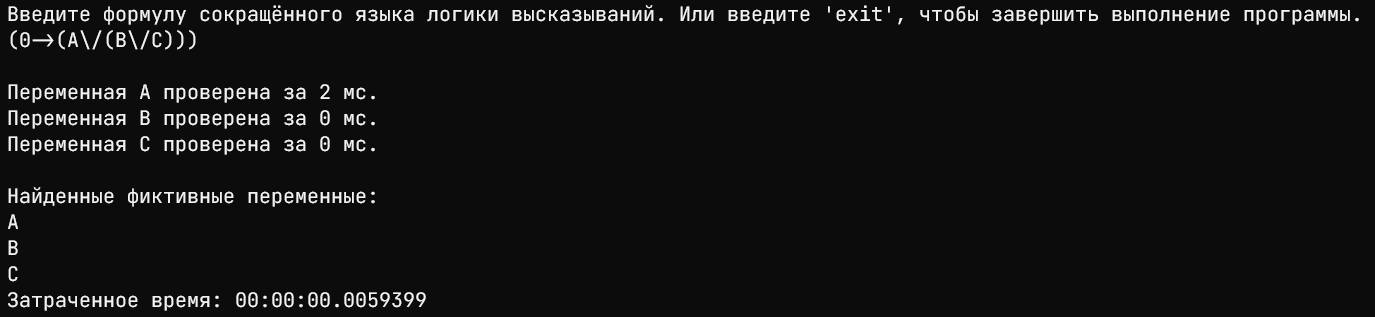


Рис. 45. Поведение программы, если пользователь ввёл корректную формулу и в ней есть фиктивные переменные

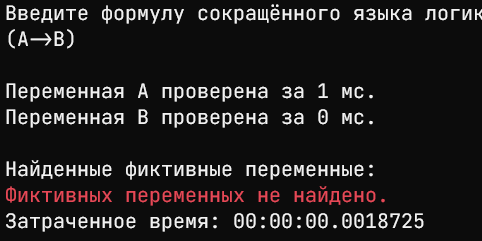


Рис. 46. Поведение программы, если пользователь ввёл корректную формулу, но фиктивные переменные в ней не обнаружены

В случае, если пользователь выбрал режим «Тестирование знаний пользователя», то пользователю будет предложено выполнить 10 заданий по нахождению количества фиктивных переменных в случайно сгенерированных формулах. После выполнения 10-ти заданий пользователь получит оценку своих знаний от 0 до 10-ти баллов.

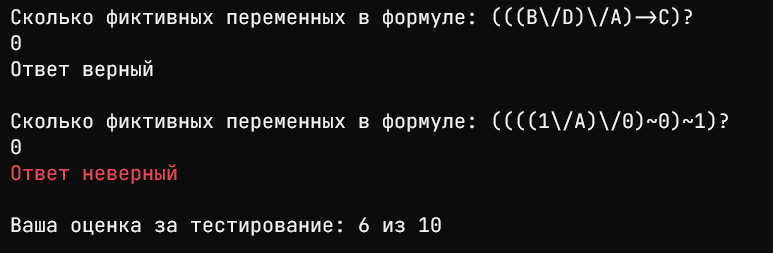


Рис. 47. Последние 2 вопроса в тесте и оценка за тестирование

**Примеры:**

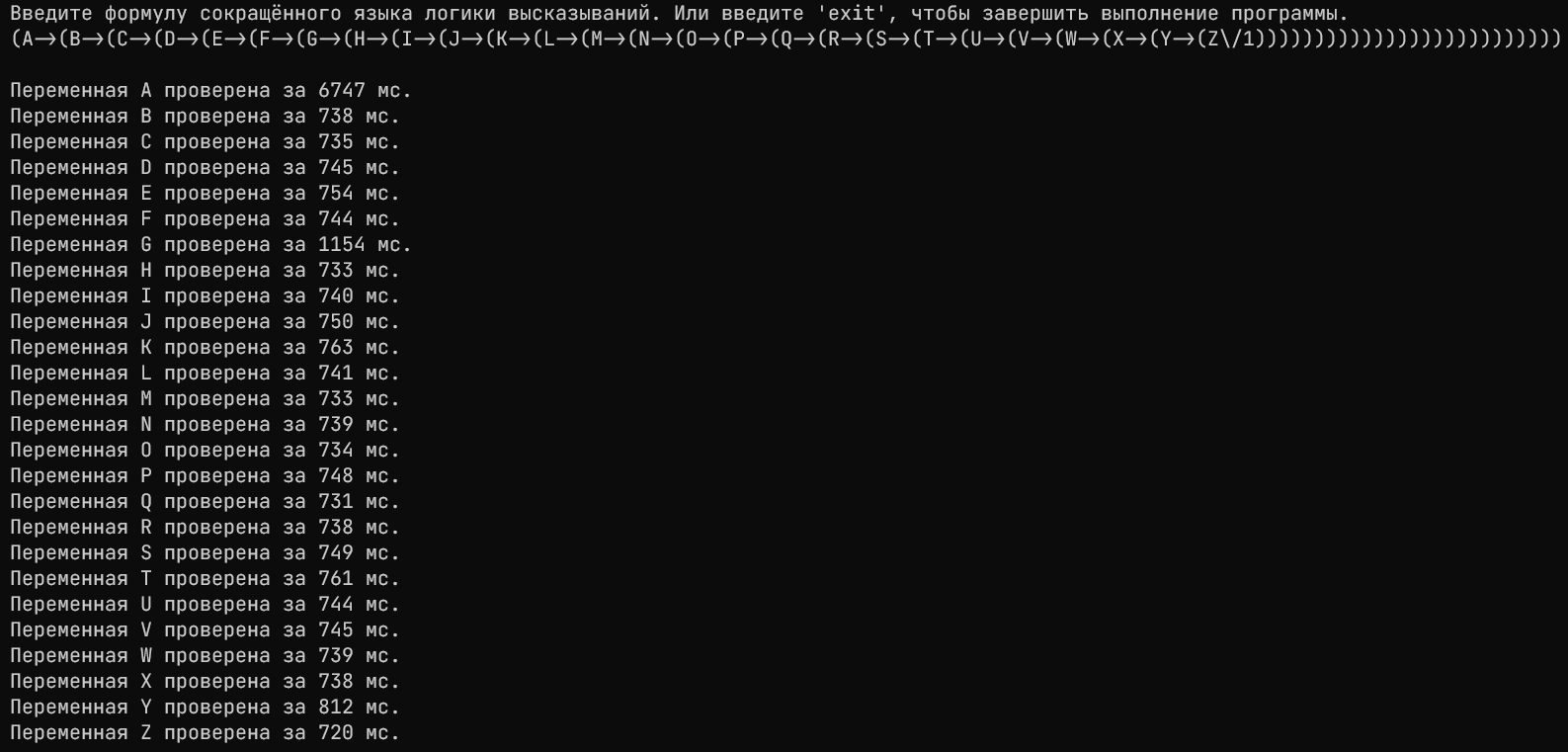


Рис. 48. Пример 1, часть 1

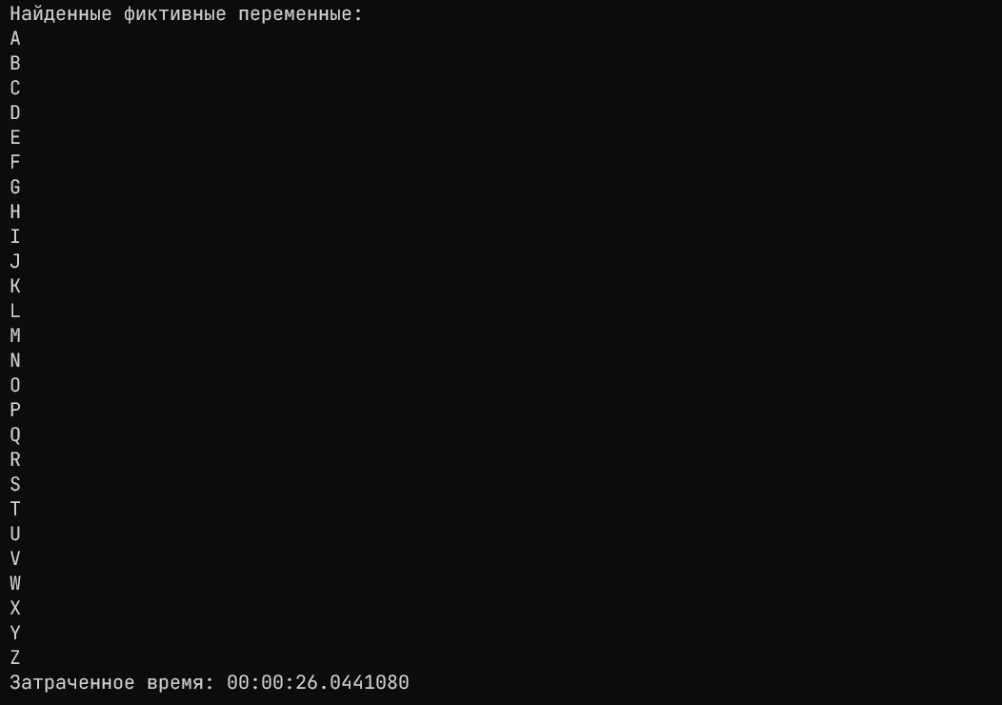
****

Рис. 49. Пример 1, часть 2

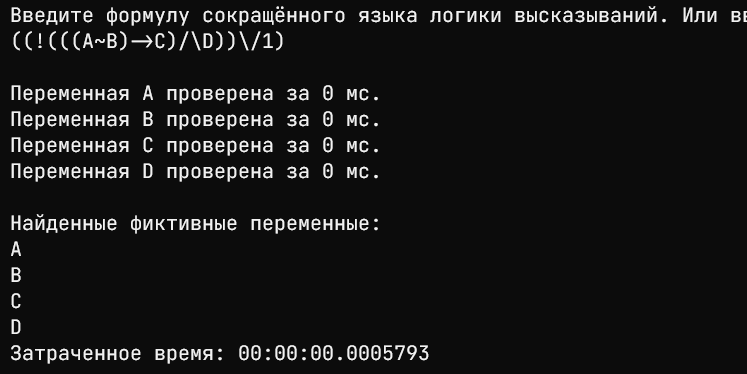
****

Рис. 50. Пример 2

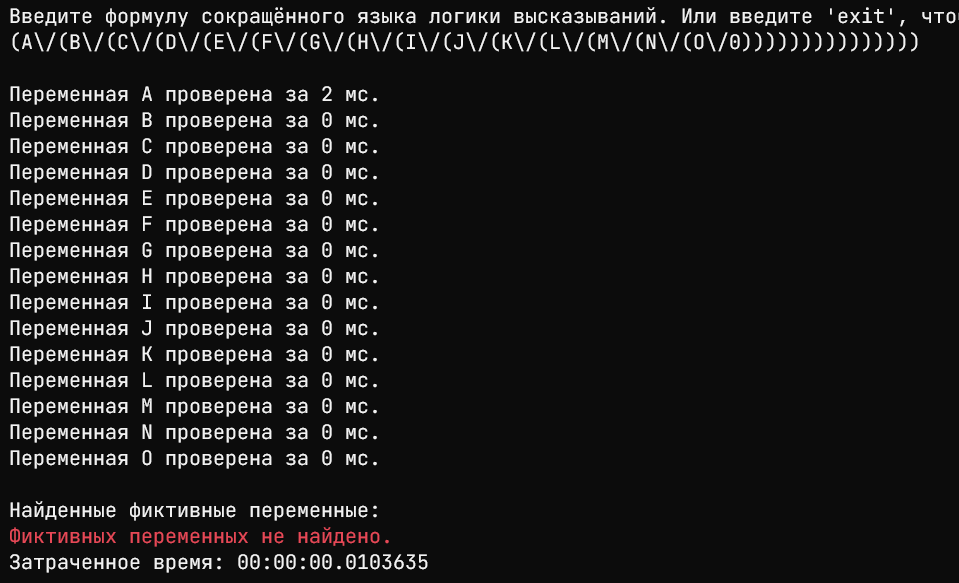
****

Рис. 51. Пример 3

**Вывод:**

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

**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.