

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 дисциплине

Логические основы интеллектуальных систем

Выполнили:

И. В. Якимович

Студент группы

121703

Проверил:

В. П. Ивашенко

Минск 2023

**Тема:** Представление и синтаксическая проверка формул языка логики высказываний

**Цель:** Получить навыки синтаксического разбора формул языка логики высказываний

**Задача:** Проверить, является ли формула ДНФ

**Описание лабораторной работы:**

В ходе лабораторной работы необходимо проверить, является ли введенная формула ДНФ.

Для этого требуется проверить:

1. Входят ли все символы формулы в алфавит языка логики высказываний.
2. Соответствует ли введенная формула правилам грамматики языка логики высказываний.
3. Выполняются ли требования, необходимые для того, чтобы формула была ДНФ

*Для выполнения данной проверки были выделены следующие подзадачи:*

1. Анализ символов, из которых состоит формула
2. Проверка скобочной последовательности формулы
3. Проверка формулы на соответствие ДНФ
  - Разбиение формул по дизъюнкции на элементарные конъюнкции
  - Анализ элементов, из которых состоят элементарные конъюнкции

**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 языка логики высказываний:**

$\langle \text{логическая константа} \rangle ::= 1|0$

$\langle \text{латинская заглавная буква} \rangle ::= A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z$

$\langle \text{формула} \rangle ::= \langle \text{логическая константа} \rangle \mid \langle \text{латинская заглавная буква} \rangle \mid$

$\langle \text{унарная сложная формула} \rangle \mid \langle \text{бинарная сложная формула} \rangle$

$\langle \text{формула} \rangle \langle \text{закрывающая скобка} \rangle$

$\langle \text{открывающая скобка} \rangle ::= ($

$\langle \text{отрицание} \rangle ::= !$

$\langle \text{закрывающая скобка} \rangle ::= )$

$\langle \text{бинарная сложная формула} \rangle ::= \langle \text{открывающая скобка} \rangle \langle \text{формула} \rangle$

$\langle \text{бинарная связка} \rangle \langle \text{формула} \rangle \langle \text{закрывающая скобка} \rangle$

$\langle \text{бинарная связка} \rangle ::= \langle \text{конъюнкция} \rangle \mid \langle \text{дизъюнкция} \rangle \mid \langle \text{импликация} \rangle \mid \langle \text{эквиваленция} \rangle$

$\langle \text{конъюнкция} \rangle ::= /$

$\langle \text{дизъюнкция} \rangle ::=$

$\langle \text{импликация} \rangle ::= = ->$

$\langle \text{эквиваленция} \rangle ::=$

*Подформула языка логики высказываний* - формула языка логики высказываний, которая является подстрокой формулы языка логики высказываний

*Дизъюнктивная нормальная форма (ДНФ)* - это один из способов представления булевой функции в виде логического выражения, который заключается в представлении функции в виде дизъюнкции элементарных конъюнкций (слагаемых). Форма ДНФ удовлетворяет следующим условиям:

- каждое слагаемое представляет собой конъюнкцию переменных и их отрицаний;
- все переменные, от которых зависит булева функция, содержатся в каждом слагаемом;
- каждое слагаемое является уникальным, то есть не существует двух одинаковых слагаемых.

*Конституента* - элементарная конъюнкция, в которую по одному разу входит каждая переменная.

*Примеры ДНФ:*

$((A \wedge B) \vee (C \wedge D))$

$(A \vee B)$

$((A \vee B) \vee (C \wedge D))$

$(\neg B)$

*Примеры не СДНФ:*

a

$((A \wedge B) \wedge (C \wedge D))$

$(A \wedge B) \vee (A \wedge C)$

$(A \wedge B \wedge C) \vee (A \wedge C)$

**Описание алгоритма:**

1. В программе описан класс DnfChecker, в котором реализованы основной метод программы Result. Данный метод состоит из следующих шагов:

(a) NewdnfLexer() - на данном шаге производится построение лексического анализатора, который принимает на вход формулу и разбивает ее на отдельные значимые единицы.

- LITERAL: VAR | (OPB NOT VAR CLB);
- VAR: [A-Z];
- OR: '\ /';
- AND: '/ \';
- NOT: '!';
- OPB: '(';
- CLB: ')';

(b) NewdnfParser() - шаг, который производит грамматическую проверку формулы.

Производится синтаксический парсинг используя такие правила, как:

- dnf: normal\_disjunct EOF;
- normal\_disjunct: ( conjunct | (OPB conjunct OR normal\_disjunct CLB) | (OPB normal\_disjunct OR conjunct CLB) | (OPB conjunct OR conjunct CLB) );

- conjunct: LITERAL | (OPB conjunct AND LITERAL CLB) | (OPB LITERAL AND conjunct CLB);

(с) Dnf() класса DnfParser - шаг, который проверяет правило dnf и заполняет объекты lexerErrors, parserErrors соответствующими ошибками, если в процессе выполнения они возникают

2. checkIsDnf() - финальный шаг программы, проверяющий lexerErrors, parserErrors на наличие ошибок.

Схемы использованных алгоритмов:

1. Result()



Рис.1.1, метод построение лексического анализатора из формулы NewdnfLexer()



Рис.1.2, метод построение синтаксического анализатора из формулы `NewdnfParser()`



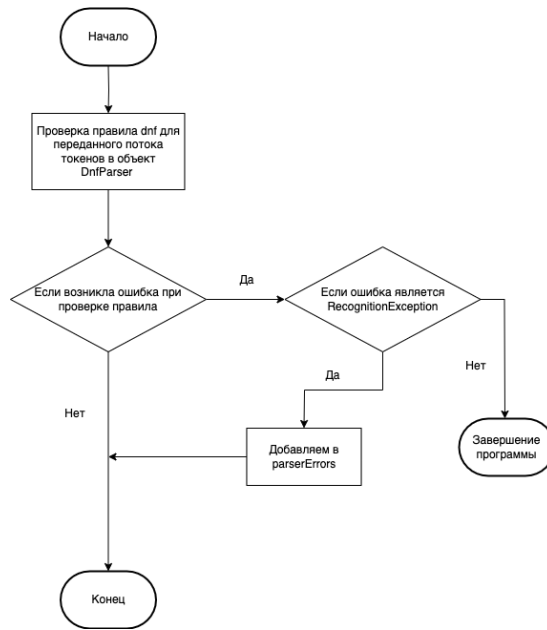


Рис.1.3, метод проверки правила dnf на синтаксическом анализаторе

## 2. checkIsDnf()

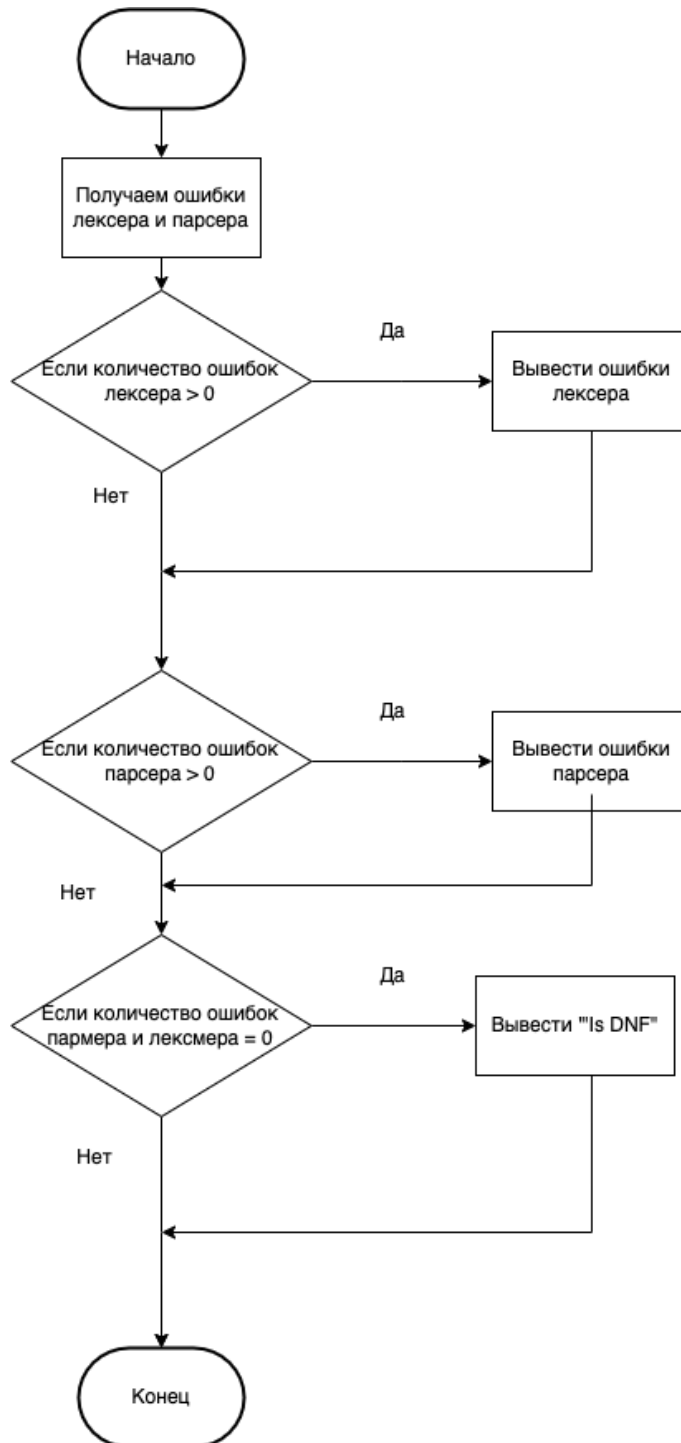


Рис. 2, метод финального решения является ли формула ДНФ или нет

## Тесты:

Для проверки работоспособности программы были проведены тесты, которые указаны ниже:

```
8 func TestIsDnf(t *testing.T) {
9     formulas := []string{
10         "(A\\B)\\/(C\\D)",
11         "(A\\B)",
12         "(A\\B)\\/(C\\D)",
13         "(A\\B)\\/(C\\D)",
14         "(A\\B)",
15         "(A\\B)\\/(C)",
16         "(A\\B)\\/(D)",
17         "(A\\B)/\\D)",
18         "(B\\((A\\B)/\\D))",
19         "(B\\/(A\\B)/\\D))",
20         "(!B)",
21         "(A\\(!B))",
22     }
23     for _, formula := range formulas {
24         checker := app.NewDnfChecker(formula)
25         lexerErrors, parserErrors := checker.Result()
26
27         AssertEqual(t, len(lexerErrors), 0)
28         AssertEqual(t, len(parserErrors), 0)
29     }
30 }
```

```
=== RUN    TestIsDnf
--- PASS: TestIsDnf (0.00s)
PASS
ok        lab/tests/unit  0.327s
```

Рис.8, результаты тестов

Все тесты, что были ранее приведены, отрабатывают правильно

### **Вывод:**

В процессе выполнения лабораторной работы были получены навыки синтаксического анализа формул языка логики высказываний с использованием фреймворка ANTLR4. Также были созданы схемы для использованных алгоритмов, выполнена отладка программы, написаны модульные тесты и проведено ручное тестирование готовой программы.

### **Список используемых 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.