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**

Выполнили студенты гр. 121702 Заломов Р.А.

Готин И.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.

**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.