МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Кафедра систем штучного інтелекту

Лабораторна робота №1

з дисципліни «Дискретна математика»

Виконав:
студент групи КН-115
Сирватка Максим
Викладач:
Мельникова Н.І.

Львів – 2019 р.

Тема: Моделювання основних логічних операцій;
Мета: Ознайомитись на практиці із основними поняттями математичної логіки, навчитись будувати складні висловлювання за допомогою логічних операцій та знаходити їхні істинностні значення таблицями істинності, використовувати закони алгебри логіки, освоїти методи доведень.

Теоретичні відомості

Просте висловлювання (атомарна формула, атом) – це розповідне речення, про яке можна сказати, що воно істинне або хибне, але не те й інше водночас.
Складне висловлювання – це висловлювання, побудоване з простих за допомогою логічних операцій (логічних зв’язок).
Найчастіше вживаними операціями є 6: заперечення, диз’юнкція, імплікація, альтернативне «або», еквівалентність.
Тавтологія – формула, що виконується у всіх інтерпретаціях (тотожно істинна формула).
Протиріччя – формула, що не виконується у жодній інтерпретації (тотожно хибна формула).
Формулу називають **нейтральною**, якщо вона не є ні тавтологією, ні протиріччям.
Закони логіки висловлювань — рівносильні, тотожно-істинні формули, що входять до структури класичної символічної логіки як формальної системи. До них **належать:** закон *тотожності*, закон *несуперечності,* закон *виключеного третього*, закон *асоціативності*, закон *дистрибутивності*, закон *ідемпотентності*, закон *комутативності*, закон *поглинання*, закон *подвійного заперечення*, закони *де Моргана* та інші.
Виконана формула – це формула, що не є протиріччям.
Предикат – це твердження, яке містить змінні та приймає значення істини чи фальші залежно від значень змінних.
Квантор - логічний оператор, що перетворює будь-який предикат на предикат меншої місності, зв'язуючи деякі змінні початкового предиката.
Вживаються два квантори: узагальнення (універсальний) та приналежності (екзистенціальний).

Існує кілька **методів доведення істинності висловлювання виду (P->Q)**

1. **Пряме міркування** (допускаємо, що висловлювання P істинне та показуємо справедливість Q);
2. **Обернене міркування** (допускаємо, що висловлювання P хибне та показуємо хибність Q);
3. Метод **“від протилежного” або метод відшукання контрприкладу** (допускаємо, що P істинне, а Q - хибне та показуємо, що все висловлювання є протиріччям);

4. Принцип **математичної індукції** - це така **теорема**:

Теорема. Нехай P(n) - предикат, який визначений для всіх натуральних n. Припустимо, що :

1) P(1) - істинне;

2) $\forall k \geq 1$ імплікація $(P(k) \rightarrow P(k+1))$ є вірною.

Тоді P(n) істинне при будь - якому n.

Перехід від P(x) до $\forall$ x P(x) або $\exists$ x P(x) називається **зв’язуванням** змінної x, а сама змінна - **зв’язаною**. Незв’язану змінну називають **вільною**.

Закони логіки першого порядку:

1. $\neg\forall x\, P(x) \equiv \exists x \neg P(x)$
2. $\neg\exists x\, P(x) \equiv \forall x \neg P(x)$
3. $\forall x(\, P(x) \wedge Q(x)) \equiv \exists x P(x) \vee \exists x Q(x)$
4. $\forall x P(x) \vee Q \equiv \forall x(P(x) \vee Q)$
5. $\exists x P(x) \vee Q \equiv \exists x(P(x) \vee Q)$
6. $\exists x P(x) \wedge Q \equiv \exists x(P(x) \wedge Q)$
7. $\forall x P(x) \rightarrow Q \equiv \exists x(P(x) \rightarrow Q)$
8. $\exists x P(x) \rightarrow Q \equiv \forall x(P(x) \rightarrow Q)$
9. $\forall x \forall y P(x,y) \equiv \forall y \forall x P(x,y)$
10. $\exists x \exists y P(x,y) \equiv \exists y \exists x P(x,y)$
11. $\forall x P(x) \equiv \forall t P(t)$
12. $\exists x P(x) \equiv \exists t P(t)$

Закони логіки висловлювань:

1.	$A \vee B = B \vee A$	$A \wedge B = B \wedge A$	Комутативність
2.	$A \vee (B \vee C) = (A \vee B) \vee C$	$A \wedge (B \wedge C) = (A \wedge B) \wedge C$	Асоціативність
3.	$A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C)$	$A \wedge (B \vee C) = (A \wedge B) \vee (A \wedge C)$	Дистрибутивність
4.	$A \vee \overline{A} = 1$	$A \wedge \overline{A} = 0$	Комплементність
5.	$\overline{A \vee B} = \overline{A} \wedge \overline{B}$	$\overline{A \wedge B} = \overline{A} \vee \overline{B}$	Закони де Моргана
6.	$A \vee (A \wedge B) = A$	$A \wedge (A \vee B) = A$	Закони поглинання
7.	$A \vee (\overline{A} \wedge B) = A \vee B$	$A \wedge (\overline{A} \vee B) = A \wedge B$	Блейка-Порецького
8.	$A \vee A = A$	$A \wedge A = A$	Ідемпотентність
9.	$\overline{\overline{A}} = A$		Інволютивність заперечення
10.	$A \vee 0 = A$ $A \vee 1 = 1$ $\overline{0} = 1$	$A \wedge 1 = A$ $A \wedge 0 = 0$ $\overline{1} = 0$	Властивості констант
11.	$(A \vee B) \wedge (\overline{A} \vee B) = B$	$(A \wedge B) \vee (\overline{A} \wedge B) = B$	Склеювання
12.	$A \Rightarrow B = \overline{B} \Rightarrow \overline{A}$		Закон контрапозиції

Випереджена нормальна форма – формула, записана у вигляді $Q_1x_1Q_2x_2...Q_nx_nM$, де кожне Q_ix_i (i = 1,2,...,n) – це "$\forall x_i$ або $\exists x_i$, а формула M не містить кванторів. Вираз $Q_1x_1...Q_nx_n$ називають *префіксом*, а M – *матрицею формули*, записаної у випередженій нормальній формі.

Для перетворення виразів довільної форми у ВНФ необхідно виконати такі **етапи перетворення**:

1. Виключити логічні зв'язки еквіваленції (~) та імплікації (-), виразивши їх через операції диз'юнкції, кон'юнкції і заперечення;
2. Опустити знаки операцій заперечення, використовуючи закон подвійного заперечення і закони де Моргана, у тому числі для кванторів;
3. Винести квантори на початок формули, використовуючи відповідні закони, для одержання випередженої нормальної форми.

Завдання лабораторної роботи

1. **Формалізувати речення.** Якщо не можеш зробити якісно роботу, то вважай що тобі не запропонують вдалу вакансію.

Розв'язання:
P - могти зробити якісно роботу;
Q - запропонувати вдалу вакансію;
x - ти;
y - компанія;
$\neg P(x) \to \neg Q(x, y)$;

2. Побудувати таблицю істинності для висловлювань:
$(x \wedge (y \wedge z) \to (x \vee y \vee z)$

Розв'язання:
Побудуємо таблицю істинності:

x	y	z	$y \wedge z$	$x \wedge (y \wedge z)$	$x \vee y$	$x \vee y \vee z$	$(x \wedge (y \wedge z) \to (x \vee y \vee z)$
0	0	0	0	0	0	0	1
0	0	1	0	0	0	1	1
0	1	0	0	0	1	1	1
0	1	1	1	0	1	1	1
1	0	0	0	0	1	1	1
1	0	1	0	0	1	1	1
1	1	0	0	0	1	1	1
1	1	1	1	1	1	1	1

3. Побудовою таблиць істинності вияснити, чи висловлювання є тавтологією або протиріччям:

$\overline{((p \wedge q)} \vee (\overline{q} \wedge r)) \vee \overline{(\overline{p} \rightarrow r)}$.

Розв’язання:

За допомогою еквівалентних перетворень спростимо дане висловлювання:

$\overline{((p \wedge q)} \vee (\overline{q} \wedge r)) \vee \overline{(\overline{p} \rightarrow r)} = ((\overline{p} \vee \overline{q}) \vee (\overline{q} \wedge r)) \vee \overline{(\overline{p} \rightarrow r)}$

Позначимо X = $((\overline{p} \vee \overline{q}) \vee (\overline{q} \wedge r))$;Y = $\overline{(\overline{p} \rightarrow r)}$

p	q	r	$\overline{p}$	$\overline{q}$	$(\overline{p} \vee \overline{q})$	$(\overline{q} \wedge r)$	X	$(\overline{p} \rightarrow r)$	Y	$X \vee Y$
0	0	0	1	1	1	0	1	0	1	1
0	0	1	1	1	1	1	1	1	0	1
0	1	0	1	0	1	0	1	0	1	1
0	1	1	1	0	1	1	1	1	0	1
1	0	0	0	1	1	0	1	1	0	1
1	0	1	0	1	1	1	1	1	0	1
1	1	0	0	0	0	0	0	1	0	0
1	1	1	0	0	0	0	0	1	0	0

Дане висловлювання є нейтральним, оскільки воно приймає як значення 0, так і значення 1.

4. За означенням без побудови таблиць істинності та виконання еквівалентних перетворень перевірити, чи є тавтологією висловлювання:

$(((\overline{p} \rightarrow \overline{q}) \rightarrow p) \wedge ((\neg(p \rightarrow q)) \rightarrow r)) \rightarrow (p \rightarrow q)$.

Розв’язання:

Припустимо, що дане висловлювання не є тавтологією. Оскільки остання операція - імплікація, то для того, щоб висловлювання було хибним, потрібно, щоб ліва частина була вірною, а права - хибною:

$((\overline{p} \rightarrow \overline{q}) \rightarrow p) \wedge ((\neg(p \rightarrow q)) \rightarrow r) = T$

(1.1)

$p \rightarrow q = F$

(1.2)

Висловлювання $p \rightarrow q = F$ є хибним лише тоді, коли p = T, a q = F. Підставляємо дані значення у рівність (1.1):

$((F \rightarrow T) \rightarrow T) \wedge ((\neg(T \rightarrow F)) \rightarrow r) = T$

$(T \rightarrow T) \wedge (T \rightarrow r) = T$

$T \wedge (T \rightarrow r) = T$

(1.3)

Оскільки висловлювання (1.3) вірне і останньою операцією є кон’юнкція, то $T \rightarrow r = T$, тобто r повинне бути вірним.

Отже, при *p = T, r = T, q = F* задана формула набуває значення F, тобто **не є тавтологією.**

5. Довести, що формули еквівалентні:

$(\overline{q} \wedge r) \rightarrow p$ і $p \rightarrow (q \wedge r)$.

Розв'язання:

Позначимо $S = (\overline{q} \wedge r) \rightarrow p; Z = p \rightarrow (q \wedge r)$.

Побудуємо таблицю істинності для цих формул:

q	r	p	$\overline{q}$	$\overline{q} \wedge r$	$q \wedge r$	S	Z
0	0	0	1	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	1	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	0	0	1	1
1	0	1	0	0	0	1	0
1	1	0	0	0	1	1	1
1	1	1	0	0	1	1	1

Оскільки два останні стовпці не індентичні, то дані висловлювання **не є еквівалентними.**

Завдання 2

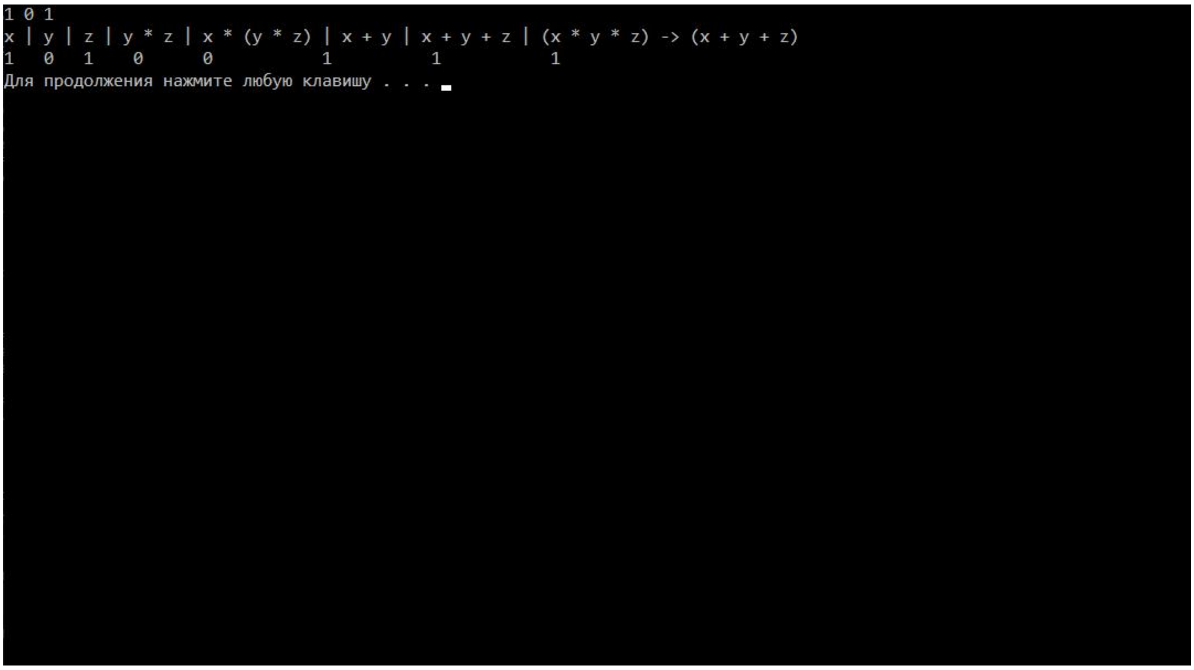
Написати на будь-якій відомій студентові мові програмування програму для реалізації програмного визначення значень таблиці істинності логічних висловлювань при різних інтерпретаціях, для наступної формули:

$(x \wedge (y \wedge z)) \rightarrow (x \vee y \vee z);$

Програма буде мати такий вигляд:

```cpp
#include <iostream>
#include <locale.h>

using namespace std;

int main()
{
    bool x, y, z; // оголошення змінних x, y, та z;
    cin >> x >> y >> z; // введення даних зміних вручну з клавіатури;
    int P1 = y && z; // оголошення та ініціалізація кожної логічної операції;
    int P2 = x && P1;
    int Q1 = x || y;
    int Q2 = Q1 || z;
    setlocale(LC_ALL, "Ukrainian");

    bool S;

    // визначення операції імплікації при різних вхідних даних;
    if ((x == 0 || x == 1) && (y == 0 || y == 1) && (z == 0 || z == 1))
    {
        if (!P2 && !Q2) S = true;
        else if (!P2 && Q2) S = true;
        else if (P2 && !Q2) S = false;
        else if (P2 && Q2) S = true;
        cout << "x | y | z | y * z | x * (y * z) | x + y | x + y + z | (x * y * z) -> (x + y + z)" << endl;
        cout << x << " " << "  " << y << "   " << z << "    " << P1 << "       " << P2 << "            " << Q1 << "           " << Q2 << "           " << S << endl;
        system("pause");
        return 0;
    }
    else
    {
        cout << "ПОМИЛКА!!!" << endl;
        system("pause");
        return 0;
    }

    system("pause");
    return 0;
}
```

Результат виконання програми при правильних вхідних даних:



Результат виконання програми при неправильних вхідних даних: